

DECLARATION

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: I.P.A.Gunasekara

Registration Number: 2014/MIT/017

Index Number: 14550176

Signature:

Date:

This is to certify that this thesis is based on the work of Mr. I. P. A. Gunasekara under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: Dr. D. A. S. Atukorale

Signature:

Date:

ABSTRACT

Sri Lanka is an agricultural country with paddy being the major crop. Based on the climatic conditions and rainfall experienced by the country during the year, paddy cultivation is done in two well identified seasons, namely MAHA and YALA season. The cultivation for these seasons is done using the water from reservoirs as well as that received from rainfall. Reservoir capacity, initial storage, rainfall, crop variety, start date of cultivation and management practices, etc. are important parameters for a successful cultivation under the reservoirs. One of the threats for the paddy cultivation is the opportunity cost of the water by other financially influential industries.

In arriving at management decisions, it is extremely important to have a prior knowledge on water requirements for a given crop in a given season. This can either be from the experience or through the mathematical modelling. However, the latter is more scientific and influential. The advantage of a software is that it is easily possible for the relevant professionals to test the water requirements under different scenarios and utilize the water resources efficiently among the different sectors.

The proposed software is designed to have three major data categories namely, Reservoir specific data, Crop data and the Climate data. The outputs will be presented with identification of crop, season and reservoir. Further, flexibility is given to check the results with user selected variations for the historical rainfall figures which are used in reservoir operation system. The software is coded using PHP language. Data is stored in a MySQL database.

A flexible system is designed by allowing the user to select options as well as giving him freedom to input the parameters decided by the stakeholders in practice. With the above inputs, user can run the model for any selected crop file and reservoir file and get the output.

This software will be an attractive solution for design engineers as well as the Engineers in the top management who are involved in decision making on Mahaweli water allocation.

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of this project will last longer only if the due appreciation is given to the people who frankly wish the success beyond boundaries specially with constant guidance and encouragement.

At this event of completion of the Project Thesis of the MIT Degree, it is my privileged to appreciate Dr. Ajantha Athukorale, my project supervisor. He is the key personnel behind this achievement. At the first instance, he encouraged me to undertake a reservoir related project which is my field of profession. From there, his guidance and the targets he set for me made the mechanism for the successful completion of the project.

Next I should thankful to all the MIT lecturers who spent their time to make us more knowledgeable on the field of IT. This is special in my case as I have no prior experience or qualification on IT. If not their support and flexibility, definitely I would not be able to write this page. Their corporation is strong enough to overcome many issues.

Eng. S.Shanmugasivanathan, retired Director of Irrigation (Training), is the personnel who helped and encouraged me to enroll and complete this programme. It is my pleasure to be thankful to him for his understanding and the unconditional support.

My good friends I found during the MIT degree programme is the other category that I should mention in this acknowledgement. All of us spent happy hours to days to years by sharing our problems and knowledge during this two-years. Our friendship is enthusiastic enough to forget many problems in personnel and professional life occurred due to the tough academic activities.

TABLE OF CONTENTS

ABSTRACT.....	II
ACKNOWLEDGEMENT.....	III
TABLE OF CONTENTS.....	IV
LIST OF FIGURES.....	VIII
LIST OF TABLES.....	XI
LIST OF ACRONYMS AND PRINCIPAL SYMBOLS	XII
1 INTRODUCTION	1
1.1 Motivation	1
1.2 Problem Domain and its Current status.....	2
1.3 Objective	4
1.4 Project Scope.....	5
1.5 Limitations.....	5
1.5.1 Mixing of the crop.....	6
1.5.2 Time unit	6
1.5.3 Reservoir combinations.....	6
1.5.4 Editing the Crop or Reservoir data	7
1.5.5 Presentation of results	7
1.5.6 Online access.....	7
1.6 Structure of the Thesis.....	7
2 BACKGROUND AND LITERATURE REVIEW	9
2.1 Proposed solution	9
2.2 Literature Review	9
2.2.1 Design of Irrigation Headworks.....	9
2.2.2 Guidelines for computing crop water requirements.....	10
2.2.3 Crop yield response to water.....	10

2.3	Similar Systems	11
2.3.1	CROPWAT 8.0.....	11
2.3.2	CRIWAR 3.0	13
2.3.3	CropWaterUse	14
2.4	Reservoir Operation technology	14
3	ANALYSIS AND DESIGN.....	15
3.1	Analysis	15
3.1.1	Requirement Gathering.....	15
3.1.2	Functional Requirements	16
3.1.2.1	System Use case diagram	18
3.1.3	Non-Functional Requirements.....	19
3.2	Design	20
3.2.1	Client/Server Architecture	20
3.2.2	Object-Oriented Architectural Style	22
3.2.3	Design Methodology	23
3.2.4	Class diagram of the system	25
3.2.5	26
3.2.6	Sequence diagrams	26
3.2.7	ER diagram	27
3.2.8	User Interfaces	29
3.2.9	Database.....	34
4	IMPLEMENTATION	35
4.1	Technical Background	35
4.1.1	Terms and Definitions	35
4.1.2	Water requirement for the Cultivation.....	39
4.1.3	Crop Water Requirement.....	40

4.1.4	Climatic and Seasonal influence	42
4.1.5	User Flexibility	43
4.1.6	Calculation Flowcharts	43
4.2	Software Implementation	46
4.2.1	Sample code	46
5	EVALUATION AND TESTING	47
5.1	Testing	47
5.1.1	Functional Testing.....	47
5.1.2	Unit Testing.....	47
5.1.3	Integration Testing	48
5.1.4	System Testing	48
5.1.5	Acceptance Testing	48
5.2	Test Plan and Test Cases	49
5.2.1	Test case for failure to save the Reservoir-Season data without a Required field... 49	
5.2.2	Test case for Retrieving data and saving the Reservoir-Season file	50
5.2.3	Test case for valid user logging	51
6	CONCLUSION AND FUTURE WORK	53
6.1	Difficulties Encountered and Achievements	53
6.2	Usefulness	54
6.3	Future work	54
	REFERENCES.....	55
	APPENDIX A USER MANUAL	56
A 1	Welcome page	56
A 2	Logging page.....	56
A 3	user with successful logging.....	57
A 4	view data.....	58

A 5	Add Data	62
A 6	delete data	65
A 7	edit data.....	67
A 8	operation execution.....	68
A 9	user registration.....	71
A 10	remove registered user	73
A 11	password reset.....	74
APPENDIX B	DATABASE TABLE DETAILS.....	75
APPENDIX C	– TEST CASES.....	80
C 1	- Test case for invalid user logging	80
C 2	Summary list of Test Cases.....	80
APPENDIX D	– SAMPLE CODES.....	84
D 1	sample code for retrieving the data from database & save in table ‘save’	84
APPENDIX E	- - CALCULATION FLOW CHART	99
E1	flow chart for calculating the crop water requirement.....	99
APPENDIX F	- RESERVOIR SYSTEM LAYOUT	101
APPENDIX G	– INITIAL DOCUMENTS	102
G 1	conceptual user interfaces	102
APPENDIX H	– USER CERTIFICATES	104
APPENDIX I	– FINAL RESULTS	105

LIST OF FIGURES

Figure 2-1 screen view of the user interface for rainfall data input of CROPWAT 8.0	12
Figure 3-1 System high level use case diagram	18
Figure 3-2 System Environment	21
Figure 3-3 Illustration of the adopted development methodology.....	23
Figure 3-4 Class diagram of the system.....	25
Figure 3-5 sequence diagram for execution of operation	27
Figure 3-6 sequence diagram for saving the reservoir-season file.....	27
Figure 3-7 ER diagram for the reservoir operation system.....	28
Figure 3-8 Welcome page of the system.....	29
Figure 3-9 Logging interface	29
Figure 3-10 main menu after admin logging	30
Figure 3-11 view data page	30
Figure 3-12 Main interface once logged as level 2 user	31
Figure 3-13 Main interface once logged as level 3 user	31
Figure 3-14 Main interface once logged as level 4 user	32
Figure 3-15 Interface for adding reservoir-season file.....	32
Figure 3-16 Interface for adding crop-season file.....	33
Figure 3-17 Final results of the programme.....	33
Figure 3-18 Database record of saving the data through reservoir-season interface	34
Figure 4-1 Elevation-Area_Capacity diagram for Giritale reservoir	36
Figure 4-2 Typical cross section of a reservoir showing important parameters	36
Figure 4-3 Selected reservoirs on the Argo-ecological regional map of Sri Lanka.....	38
Figure 4-4 variation of the crop factor with growth stage	40
Figure 4-5 Factors affecting on the CWR.....	41
Figure 4-6 Illustration of the independent overlapping of the crop factor with months.....	42
Figure 4-7 Flowchart for calculating the monthly inflow due to rainfall	44
Figure 4-8 Calculation of the final storage of the reservoir.....	45
Figure A-1 Welcome page of Reservoir Operating System	56
Figure A-2 user logging page.....	56
Figure A-3 Main page of the system for level 1 user or Admin	57

Figure A-4 Main page of the system for level 2 user	57
Figure A-5 Main page of the system for level 3 user	58
Figure A-6 Main page of the system for level 4 user	58
Figure A-7 Interface facilitates the user to select any option to view all categories of data	59
Figure A-8 interface to view the rainfall data.....	60
Figure A-9 interface for viewing the crop data	60
Figure A-10 interface for viewing the reservoir-season data	61
Figure A-11 interface for viewing the crop-season data	61
Figure A -12 Main page to add data	62
Figure A-13 Interface for adding the rainfall data.....	63
Figure A-14 Interface for adding the crop data, trying to save with required fields empty	63
Figure A-15 Response when adding invalid data through reservoir-season interface	64
Figure A-16 Interface for adding the reservoir data	64
Figure A-17 interface for adding the seasonal specific yield data	65
Figure A-18 Main page to delete the data	66
Figure A-19 Page to delete the rainfall data	66
Figure A-20 Alert message before execute the deletion.....	66
Figure A-21 system response message after deletion.....	67
Figure A- 22 Main page to edit the data.....	67
Figure A-23 Interface for edit and save the reservoir-season data.....	68
Figure A-24 Interface for execute the operation	69
Figure A-25 Presentation of results after executing the programme.....	69
Figure A-26 Graphical presentation of key parameters of the final output.....	70
Figure A-27 interface for user registration	71
Figure A-28 e-mail validation in user registration	71
Figure A-29 Interface for password validation	72
Figure A-30 System response after successful user registration	72
Figure A-31 Interface to select the user to be removed.....	73
Figure A-32 Alert message to request the confirmation to delete.....	73
Figure A-33 Confirmation message after deletion	73
Figure A-34 interface to change the password	74
Figure A-35 Alert message in entering wrong pass	74

Figure A-36 Confirmation message in password reset	74
Figure E -1 Flow chart for calculating the crop water requirement for entire growth stages	100
Figure F -1 Mahaweli water distribution system with selected reservoirs.	101
Figure G -1 Initial paper prototype to add reservoir-season data	102
Figure G-2 Initial paper prototype to add crop-season data.....	103
Figure I-1 Results of the Operation study in tabular form.....	105
Figure I-2 Results in graphical form.....	106

LIST OF TABLES

Table 1-1 Summary of the reservoirs to be considered in the study	3
Table 5-1 Test case for failure to save reservoir-season file without a single required field	50
Table 5-2 Test case for saving the reservoir-season data with valid data	51
Table 5-3 Test case for registered user logging.	52
Table B-1 Database table 'reservoir_data'	75
Table B-2 Database table 'area_capacity'	75
Table B-3 Database table 'crop_varity'	76
Table B-4 Database table 'evaporation'	76
Table B-5 Database table 'season'	76
Table B-6 Database table 'iso_yield'	77
Table B-7 Database table 'rainfall'	77
Table B-8 Database table 'ref_crop_evpotr'	77
Table B-9 table crop_save, where data saved from crop-season UI	78
Table B-10 table save, where data saved from reservoir-season UI	79
Table C-1 Test case for unregistered user logging.	80
Table C-2 test cases on user management	81
Table C-3 Test cases on add crop data	82
Table C-4 Test cases on add crop_season data	83

LIST OF ACRONYMS AND PRINCIPAL SYMBOLS

DoA	-	Department of Agriculture
FAO	-	Food and Agriculture Organization of the United Nations
ID	-	Irrigation Department
MASL	-	Mahaweli Authority of Sri Lanka
WMS	-	Water Management Secretariat
WMU	-	Water Management Unit
FSL	-	Full Supply Level
MOL	-	Minimum Operating Level
ET ₀	-	Reference crop evapotranspiration of the respective month
ET _c	-	Crop water requirement
ET ₀₁	-	ET ₀ for 1 st month within which the part of the crop growth stage concerned falls
ET ₀₂	-	ET ₀ for 2 nd month within which the part of the crop growth stage concerned falls
K _c	-	Crop coefficient which varies for different growth stages.
AJAX	-	Asynchronous JavaScript and XML
ER	-	Entity Relationship
GUI	-	Graphical User Interface
HTML	-	Hyper Text Mark-up Language
IT	-	Information Technology
JS	-	JavaScript
PHP	-	Hypertext Preprocessor
RAM	-	Random Access Memory
SQL	-	Structured Query Language
UML	-	Unified Modelling Language
WAMP	-	Windows, Apache, MySQL, PHP

1 INTRODUCTION

As an agricultural country with many reservoirs constructed primarily to deliver water for paddy cultivation, Sri Lanka has been practicing the water conservation principles historically. Paddy is the main cultivation in Sri Lanka. More than 400,000 hectares are cultivated twice a year under major reservoirs in Sri Lanka. All the reservoirs in Mahaweli systems get the water from rainfall on its own catchment as well as from the Mahaweli system. The water from the own catchment due to rainfall is not sufficient for the cultivation of two seasons. The general practice is to divert water from Mahaweli system when there is a shortage of water. However, the approval of the Mahaweli Water Panel (MWP), which is represented by the Secretaries to the respective Ministries and the Heads of the respective Departments/Boards is necessary for such diversion. Whenever such a diversion is granted, it is mostly at the cost of Hydropower. In some cases, the decisions have been taken to forgo the cultivation due to uncompromised hydropower demand.

1.1 MOTIVATION

The complex water demands in the multi sectoral water demand system urges a proper management plan as well as a forecasting mechanism. As agriculture is a prominent industry in the country, having high potential to economic and political sensitivity, government gives priority to the Agriculture needs. However, the agricultural water demand is not presented properly to the authorities who do the water allocation, by the respective government organizations. This is basically due to the lack of software facility to check the alternative solutions and arrive to an optimal solution.

Even though the calculation of the water requirement for the cultivation can be done systematically, it is not very easy to couple it with reservoir specific parameters, crop specific parameters, historical rainfall and evaporation data, commencement date of the cultivation season, and finally balance the reservoir on monthly basis without a computer application. There is no such software available commercially as well as organizationally so far in Sri Lanka. Even though it was tried by many organizations, it was not materialized basically due to complexities in the calculations, lack of text books. Due to above reasons, the domain experts were unable to explain the problem to the developers properly. Further there is no qualified IT staff in most of the organizations responsible for this issue. There are spreadsheet applications developed by individuals for their convenience.

Major drawback of these worksheets is the inability to retrieve the data from a database. Due to this reason, spreadsheet developed for a particular reservoir cannot be used for another one.

This problem can easily be solved by storing the data of different reservoirs, crops in a server database and write a programme to extract the data and do the calculations with essential user inputs.

As an Engineer I faced many difficulties in justifying the water demands for cultivation to the top level of the *Water Panel* which has the sole authority to allocate the Mahaweli water. Water Panel consists of secretaries of stakeholder ministries and the Heads of the ID, MASL, CEB. This difficulty is mainly due to the inability to present the requirements for different scenarios in a transparent and attractive format. Many of the professionals of above organizations expect the use of standard software those available in other industries.

With the above explained actual reasons and the need to help the Irrigation Department that I work I feel great encouragement to find a solution to the problem.

1.2 PROBLEM DOMAIN AND ITS CURRENT STATUS

Polonnaruwa district is one of the largest irrigation water consumer in the country. Its total extent cultivated through the Mahaweli system is about 32,260 hectares (80,000 Acres). Out of the above area, around 50,000 acres are in a single Water Management Unit. (WMU). This single WMU includes three major reservoirs, namely Minneriya reservoir, Kaudulla reservoir and Giritale reservoir. In addition to these three reservoirs Kantale reservoir in Trincomalee district is also falls to this WMU. Thus, the total area covered under the study will be around 27,420ha (68,000 Acs).

All the reservoirs considered for the study are fed by a single canal system of Mahaweli diversion. The upper part of which is called Elahera-Minneriya Yoda Ela (EMYE) and the lower part is called as Minneriya-Kantale Yoda Ela (MKYE). As the name of the canal indicates, it starts at Elahera at the Mahaweli River and ends at Kantale which goes through the Minneriya Reservoir. The water distribution network for the selected area is shown in *figure F.1* in APPENDIX F .

The capacities of these reservoirs and the respective irrigation extends are given in *table 1-1*.

Name	Capacity (MCM)	Cultivable Area (Acs.)
Minneriya reservoir	135.5	25,000
Kaudulla reservoir	127.0	15,000
Giritale reservoir	26.7	8,000
Kantale reservoir	140.61	20,000

Table 1-1 Summary of the reservoirs to be considered in the study

The farmers virtually have their own choice of selecting the crop type that are going to be cultivated during the season. The start-date Further it could differ from reservoir to reservoir.

Reservoir has its own climatic and physical characteristics as well. Capacity, FSL, MOL, Elevation-Area-Capacity relationship and the system efficiency are the most important reservoir specific parameters related to the problem.

The crop that is going to be cultivated for the season also has its own properties. Paddy plant have four different growth stages which combinedly make the lifespan of the plant. Each of these growth stages has different water requirements. Monthly water requirement is directly depending on the water consumption of the plant at different growth stages.

Other most important parameters are the climatic parameters. Evaporation and rainfall are the two major components in this category. Effect of humidity, wind-speed, temperature, and solar radiation is combinedly addressed in the evaporation. Rainfall is the other key whether parameter that could influence the water requirement of the crop as well as the inflow to the reservoir.

Final irrigation water requirement, normally termed as *irrigation demand*, is a result of a combine effect of the reservoir parameters, crop, climate and the farmers' behaviors.

Mahaweli water diversion for each reservoir can be determined only after analyzing the combine effect of above factors.

Even though the crucial issues which are described earlier, are frequent in this combined system, there is no proper method of predicting the Mahaweli diversion requirement for the irrigation schemes considered in this study. In many cases these requirements are calculated based on the

practical experience of recent years and the predicted rainfall during the forthcoming cultivation season. Sometimes the water demand will be calculated on thumb-rule basis. One of the disadvantage of the existing mechanism is that there is no scientific way to prove the requirements to the stake holders. In many cases, there are suspicions among the authorities over the historical figures presented to request the water quota. These creates a lot of issues among the people of different priorities for water. Ultimately it has become a crucial need to have a systematic approach to find out the initial water storages of the reservoirs, irrigation water needs and Mahaweli water diversions. More importantly this solution should be a user-friendly one so that it could facilitate to test different combinations discussed earlier in this report.

1.3 OBJECTIVE

The objective of the software is to determine the initial storage of individual reservoirs at the start of each cultivation season that could be adequate enough to cater the irrigation demand with the help of the Mahaweli diversions, catchment inflows and the spilling from the upstream reservoirs, while satisfying the following conditions. Here, Mahaweli diversion is a user input parameter.

- ✚ The first condition is that any reservoir shall not be spilled during the season unless otherwise its spilled water is considered for the success of the downstream reservoir.
- ✚ The second condition is that reservoir should cater its approved cultivable area.

Reservoir water balance is calculated using the simple equation “*initial storage + inflows – water issues – losses – spilling = Final Storage*” on monthly basis. All these parameters and methodology of calculating those parameters is discussed in latter chapters.

In achieving the main objective following are the key milestones that should be achieved.

- ✚ Calculate the water requirement for any variety of paddy. The crop water requirement varies with the durations of the different growth stages of crop and its growth pattern,
- ✚ Determine the monthly inflow to any reservoir based on the historical rainfall data with user modified level of variation.
- ✚ Determine the influence of rainfall that could cater the part of crop water requirement. This part of rainfall is termed as effective rainfall.

- ✚ Determine the evaporation loss and the seepage loss from the reservoir. Evaporation loss from the reservoir water surface which directly proportional to the reservoir water surface area of that month and the respective evaporation rate of the month. Seepage loss is a predetermined percentage of the existing storage.
- ✚ Calculate the water to be released from the reservoir for the successful cultivation by considering the above factors and the possible diversion from Mahaweli and any upstream reservoir. Application efficiency and the conveyance efficiency should also be included here.
- ✚ Determination of the end of the month storage and water levels of the reservoir by considering the above details. This will ultimately calculate the spilling of the reservoir during the month, which could be a waste if not utilized by the downstream reservoir. The month-end storage is the beginning storage of the next month.
- ✚ Present the monthly water usages, spilling and end of month water levels on a seasonal basis for the selected reservoir.

1.4 PROJECT SCOPE

The reservoir operation system is expected to be use as a tool of calculating the irrigation water needs of the major reservoirs in the Polonnaruwa District and the Kantale reservoir. Further, this is a smart tool of accurate & flexible decision making for the management. This will help the management to take decisions with high confidence.

The final results, reservoir water balance on monthly basis for the season, is expected to be calculated from the two files extracted from the database. One file represents the reservoir-season combined data and the other represents the crop-season combined data. Both the files will be created by the users and saved in the database in prior occasions. To create these two files user will have to select relevant data from the database together with his own inputs.

Facilities will be given to the user to add new crop varieties, new reservoirs and other related data fields. Further the user will be able to view the existing data through the interfaces. With the user's choice, user can add two main files which are necessary for the calculation of the final result.

1.5 LIMITATIONS

There are several limitations identified while formulating the project. Many of the limitations are due to the time availability to complete the work as well as to keep the system more user-friendly

comparing to the effect of the minor parameters. In real terms those limitations do not make considerable impact on the practice in the field. However, those limitations are discussed below.

1.5.1 Mixing of the crop

Very rarely, there can be more than one crop with different lifespans could be cultivated during the season under a single reservoir. Depending on the variety, the actual water requirement will be changed. But in case of paddy, this change is negligible when comparing to the actual amount of water used by the paddy. This is mainly due to the reason, that the water is supplied not according to the exact requirement, but at a fairly higher rate to pool the paddy fields as well. In fact, this difference will be less than 5% comparing to the actual usage. It is the standard practice, that government instruct farmers to cultivate paddy varieties of same duration under any given reservoir in a particular season. Hence the above limitation does not practically yield any negative impact to the final results.

However, the provision is made available to address this issue by introducing the factor “percentage cultivated” in the interface. So the user can create two files to suit the percentages of different crops and separately calculate the crop water requirements.

1.5.2 Time unit

The data can be saved through the interfaces only on seasonal basis. The rainfall, and other water consumptions are calculated on monthly basis. The final output will also be presented on monthly basis. It is not possible to run the programme on weekly or daily basis and get the results on monthly basis. In fact, this is not a practical need, as the reservoir storages and rainfall are calculated on monthly basis in Sri Lanka.

1.5.3 Reservoir combinations

The physical parameters of the reservoirs are stored in the database. In a very complex situation, one can think to do the cultivation of a paddy fields of one reservoir, under another reservoir in a cascade system. This can happen only one reservoir is breached and out of service. In the proposed solution, the area to be cultivated and the climatic properties of that area are considered as attributes of the reservoir. In the proposed solution, it is possible to increase the cultivable area by multiplying a factor, but the attributes of the area cannot be changed. In any case if such testing to be needed,

the possibility is to include an artificial record with different reservoir name. However, this is an extremely rare and remote requirement.

1.5.4 Editing the Crop or Reservoir data

The user is not allowed to edit the reservoir or crop data which are stored in the database. Only the admin can edit such data. However, the privileged user can add new crop or reservoir data in a different name. Editing is a complex situation as a single wrong figure will lead to a completely wrong answer. However, adding new data, such data could be saved as a new record.

1.5.5 Presentation of results

Results are expected to present as a combination of “reservoir-crop-season”. All the important details will be displayed in the results on monthly basis. But it is not designed to present the results, on annual basis, or compare the results of two reservoirs or two seasons. Annual basis results are not practically important. But comparison of the results in two seasons is little bit interesting.

1.5.6 Online access

At this stage, the software is only accessible through localhost. This can be easily modified to get the access through the Irrigation Department website in future.

1.6 STRUCTURE OF THE THESIS

The thesis is presented under five main topics.

Chapter 2, Background and Literature review, identifies the similar systems available in this category. Further it discusses the systems available and the limitations of those software in Sri Lankan context. The positive and negative points are discussed here. Further the reasons for limitations of similar systems identified above are also discussed.

Chapter 3, Analysis and Design, is reserved to discuss the factors relevant to the Analysis & Design of the proposed system. The architecture of the system, design considerations, are discussed and presented in a diagrammatic form as well. Aspects related to the Software requirement specification is also discussed in this chapter. Further this includes a detailed overview of functional and non-functional requirements of the project. User interfaces and the final results are included in the latter part of this section.

In chapter 4, Implementation, progress of the project is explained. The technical background of the project is explained here. Sample codes with complex flow diagrams are included here.

Chapter 5, Evaluation and Testing, discusses regarding the quality aspects, the testing and evaluation of the project. Sample test cases are discussed here.

The last chapter, chapter 6, Conclusion and Future work, discussed the future work that will improve the usefulness of the proposed system together with the concluding remarks.

2 BACKGROUND AND LITERATURE REVIEW

2.1 PROPOSED SOLUTION

Analyzing the water diversion requirement in a transparent way for the reservoirs is the solution for the issues discussed in *chapter 1.2*. Also, this will help to manage water in the Mahaweli system more economically viable and sustainable manner. Under this project study, it is expected to quantify the Mahaweli diversion requirement in a scientific method through the development of a software using PHP and MySQL. The data required for the calculation will be obtained from the publication of *Guidelines for computing crop water requirements* [1], *Crop yield response to water* [2], *Design of Irrigation Headworks* [3] and the *Watersheds of Sri Lanka* [4]. The calculations will be done in accordance with the standard reservoir operation study methodology specified by the *Design of Irrigation Headworks*.

2.2 LITERATURE REVIEW

There are plenty of text books published on Crop Water Requirement (CWR) by international organizations and professionals. Out of these publications [1] is the most widely accepted one. UN as the publisher of the book, it addresses almost all the aspects of climate and the crop as far as the cultivation is concerned. However, it does not address the cultivation issues together with the irrigation under a reservoir. This part, being specific to the country, is perfectly addressed by local publications. The main difference between their systems and our practice is that the Sri Lankan irrigation systems totally depends on the reservoirs & water delivery through canals whereas European systems depends on the closed network with pumping. The one and only the text book that guides and explained the reservoir operation for irrigation reservoirs is *Design of Irrigation Headworks* [3] published by the Irrigation Department in 1982.

There are few software applications available in calculating the crop water requirements. All of them are developed to address the water requirements under the European irrigation technologies. However, the pros & cons of three widely accepted software are discussed in this chapter.

2.2.1 Design of Irrigation Headworks

In this book, it clearly describes the factors to be considered for crop water requirements for paddy cultivation and reservoir storage variation during the season. Author of the book is the late Director

of Irrigation, equivalent to the post of present Director General of Irrigation, Eng. A. J. P. Ponnarajh.

This book covers almost all the aspects considered in paddy cultivation in Sri Lanka. The traditional cultivation practices are also considered in this book. This book is used as a reference material for Undergraduate Engineering Degree courses in Government universities of Sri Lanka.

The criteria for determination of the initial storage of a reservoir is well explained here. Objective of [3] is to elaborate the factors affecting on the determination of the reservoir parameters at the design stage. Nevertheless, it is possible to do the reverse way to determine the initial storage of a reservoir, area to be cultivated under the reservoir during the season or the Mahaweli inflows during the season, as all of those are user controllable variables, according to its fixed parameters and for a given crop & climate conditions.

By giving due consideration for above facts, *Design of Irrigation Headworks* guides all the way to calculate the *irrigation water requirement* and do the *reservoir operations* on monthly basis.

2.2.2 Guidelines for computing crop water requirements

This publication provides a precise guidance in determining crop water requirements and their application in planning, design and operation of irrigation projects. It suggests different methods which are available to derive crop water requirements. The use of four well-known methods, each requiring a different set of climatic data for determining CWR, is defined to obtain reference crop evapotranspiration (ET_o). ET_o denotes the level of evapotranspiration for different climatic conditions. To derive the evapotranspiration for a specific crop, relationships between crop evapotranspiration (ET_c) and ET_o are also given for different crops, stages of growth, length of growing season and prevailing climatic conditions. The effect of local conditions on crop water requirements is also discussed in this book. Some of the factors are, local variation in climate, soil water availability and agronomic and irrigation methods and practices. Some of the issues discussed in this publication cannot be adopted in Sri Lankan context due to lack of sophisticated instruments and technology.

2.2.3 Crop yield response to water

With the results of large number of experiments, both at the real fields as well as laboratory conditions, and technological advances achieved over the last few decades, FAO introduces a set of tools in this publication to better appraise and enhance crop yield response to water. These tools

provide the means to sharpen assessment and management capacities required to: compare the result of several water allocations plans, improve soil-moisture control-practices under rainfed conditions, optimize irrigation scheduling, sustainably intensify crop production; close the yield and water-productivity gaps, enhance strategies for increased water productivity and water savings. These tools are invaluable to water managers and planners and the Engineers and finally the all kind of stake holders.

In this publication, crop-water relationship was discussed for many crop varieties. But for this study, it is limited to the rice. This book addresses the influential factors on the rice cultivation in tropical countries, which are applicable to Sri Lankan context largely.

2.3 SIMILAR SYSTEMS

There are few international software packages available in the industry similar to the proposed software but all are developed based on the European technology and agricultural practices. One of the major drawback in all the systems is the lack of reservoir involvement which is key component in the Sri Lankan irrigation systems. In addition to that seasonal data based on the Sri Lankan cultivation seasons and our cultivation practices cannot be incorporated into those software. However, three widely used systems among them are discussed below as it is useful to develop the proposed system more effectively.

2.3.1 CROPWAT 8.0

This software has been developed by the World Food and Agriculture Organization. This can be downloaded without any subscription through the official website of FAO. FAO has done this as a part of their services to the society. They have developed the early versions of this programme in 1980s for the DOS and the latest version is **CROPWAT 8.0** [5]. This programme works on **Windows**.

CROPWAT 8.0 for Windows is a computer program for the calculation of crop water requirements and irrigation requirements based on soil, climate and crop data. In addition, the program allows the development of irrigation schedules, which shows the irrigation frequency, for different management conditions.

CROPWAT 8.0 includes standard crop and soil data that could be used when local data is not available. When local data are available, these data files can be modified and saved with a new name. Likewise, if local climatic data are not available, these can be obtained for over 5,000 stations worldwide from CLIMWAT [6], the associated climatic database. The development of

irrigation schedules in CROPWAT 8.0 is based on a daily soil-water balance using various user-defined options for water supply and irrigation management conditions. Scheme water supply is calculated according to the cropping pattern defined by the user.

CROPWAT 8.0 provides very user-friendly interfaces. The main window consists of main menu and vertical & horizontal tool bars. It provides interfaces for all areas where the parameters are needed to be input. The input files can be saved in respective folders of the data category such as CROP, RAIN, CLIMATE, etc. that are needed for the water requirement calculation. The main menu provides the facility for calculation and graphical presentation of results using the data saved earlier. Different extensions are used in saving the files in such a way that the extension identifies the data category. For example crop data is saved in CROP sub folder of the DATA folder with the extension of “.cro”. There are readily available data files included in the software, which can be used in the absence of actual data as well as for the demonstration purposes.

The main drawback of this programme under the Sri Lankan context is that there is no provision to include the reservoir data category. This is one of the major component in optimizing the reservoir water levels which is the main objective of this project. In many countries, the irrigation is done through pumping and using sprinkler systems. Because of that the reservoir part is not applicable for their analysis.

One of the data entry window is shown in the *figure 2 -1* below.

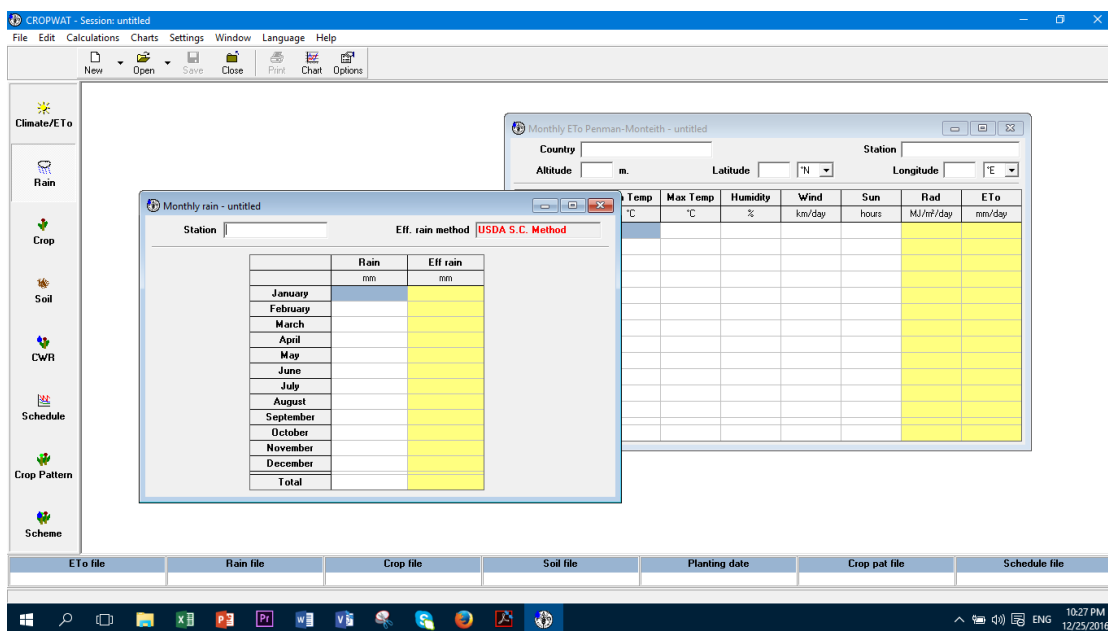


Figure 2-1 screen view of the user interface for rainfall data input of CROPWAT 8.0

All calculation procedures used in CROPWAT 8.0 are based on the FAO publications of the Irrigation and Drainage Series *Guidelines for computing crop water requirements* [1] and *Crop yield response to water* [2].

2.3.2 CRIWAR 3.0

CRIWAR 3.0 [7], a commercial one, is another software available for the calculation of crop water requirement. This is a simulation program that combines all water balance components into a single simulation procedure. This software is basically designed for the complex combinations of different climatic conditions and different crop combinations. CRIWAR 3.0 runs on Windows environment, which is developed by The Netherlands Professionals. This software is developed based on the text book *water requirements for irrigation and the environment* [8]. It too followed the FAO norms and methods as well.

CRIWAR 3.0 calculates the crop irrigation water requirements of a cropping pattern in a given irrigated area. The input data of CRIWAR 3.0 are organized through three files; *a general data file* on the irrigated area, *a meteo data file* and the *cropping pattern file*. The cropping pattern file can be composed of 50 CRIWAR programmed crops and of any user defined crop for a user-selected combination of *general data, meteo data and cropping pattern data*.

For a user-selected combination of general data, meteorological data and cropping pattern data, CRIWAR 3.0 creates tables and graphs giving; reference evapotranspiration, crop irrigation water requirements per 10-day period or month, cropping intensity, cropping pattern, effective precipitation, etc. The main advantage here is that the facility to import all tables and graphs into presentation or report writing purposes.

CRIWAR 3.0 is a useful tool in the design of irrigation systems, because it quickly calculates the summarized irrigation water requirements of a complex cropping pattern for a large area with ready-to- use output tables and graphs. This allows the design engineer to review alternative cropping patterns under various meteorological conditions.

CRIWAR 3.0 is the latest in a series of water requirement analysis and evaluation programmes developed since 1988, and hence it is well tested and developed to meet the requirements of the worldwide professionals.

The current version of the CRIWAR, CRIWAR 3.0, is maintained on the internet site www.bos-water.nl.

2.3.3 CropWaterUse

CropWaterUse [9] is the other software application discussed under literature review. This application is developed by Department of Agriculture and Fisheries, Queensland, Australia. There are no details published by the developer about the principles they used in calculating the crop water requirement. However, this application also not capable to consider the reservoir water balancing criteria.

In this software, the most advantage is that the individual user can create their own project. Under the created project, the user can modify the crop characteristics and calculate the crop water requirement.

One of the major limitation of this software is that it is not possible to add a crop on the user's choice.

This software provides a well-structured guidance, whenever the user tries to go wrong direction. According to the crop types given in the software, this is concentrated only on the Australian agriculture.

2.4 RESERVOIR OPERATION TECHNOLOGY

While operating the reservoirs for either irrigation or hydropower purpose, its storage fluctuates with time. The managers of the reservoir should update the storage data regularly. The frequency of update could depend on the purpose of the usage of the reservoir and its impact. For hydropower reservoirs, it could be weekly or even a shorter period. For the purpose of irrigation reservoirs this interval is standardized to a monthly basis. This implies that the reservoir managers can predict the water requirement for the entire month in advance and make the reservoir ready for that. In the project study, reservoir operations are considered on monthly basis as it is the practical situation. However, the reservoir operation study is done for a season, six-month period in Sri Lankan context, at once. The critical conditions are checked on monthly basis of the season whilst maintaining those requirements throughout the season. For these conditions, seasonal parameters and monthly parameters are combinedly used according to the guidelines given in the reference material [3].

3 ANALYSIS AND DESIGN

In this section, a detailed description of the Software component of *Reservoir Operation System* is presented. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This section is intended for both the users and the developers of the system.

This software is supposed to be facilitate for reservoir managers of the Irrigation Department and other higher-level officers of the Mahaweli Water Allocation System. The software has been designed to determine the reservoir water levels at the start of the cultivation season where neither the spilling nor the failure of cultivation could occur during the season. Eventually, this software will help to maximize the productivity of water by providing tools to assist in decision making and reviewing.

The problem can be sub divided into three main areas. First and most important factor is the Climate. The second one is the Physical parameters of the reservoirs including that part of Mahaweli system and the third one is the crop related parameters combining with the choice of the people. Those are discussed in detail in this chapter.

3.1 ANALYSIS

3.1.1 Requirement Gathering

Clear understanding of what the project will deliver is critical to its success. As the author is working in the Irrigation Department, it is not very much difficult to find out the requirements. However, this system is intended to use with other stakeholders of the MWP, their concerns as well as the concerns of the field level officers are also considered in developing the system.

- ✚ At the initial stage, individual interviews were done with the top management of the irrigation Department. Aim of the interview is to identify the problems they are facing in high level meetings and get their own views as the solution to the problems. Further recently retired top level officers including a former Director General were also interviewed as their views could be more impartial and independent.
- ✚ Next with the collection of their views, the field level officers, reservoir managers were interviewed to find out whether they have different views in managing their reservoirs. Especially, they have field level adjustments and practices those are not officially recorded.

- ✚ Finally, a senior officer of Mahaweli Water Management Secretariat was also interviewed. Aim of this was to get her concurrence for the proposed system and allow her to add or adjust the parameters if necessary. At this interview, a paper prototype, which was developed using the inputs of the previously explained two interviews together with the Author's experience on irrigation sector as well as the IT knowledge, was presented for her convenience rather than a questioner.
- ✚ The system contains a relational (MySQL) database, containing reservoirs, crop varieties, physical parameters of reservoirs and combined crop-climate factors. All together there are eight tables covering raw data.

In the proposed system, there will be three user groups plus the administrator. Only Admin can directly access to all components of the Database. Other user groups are identified as:

- i. Top level officers of MWMS & Irrigation Department.
- ii. Engineers of the Irrigation Department
- iii. Tertiary users of the system who only can view the information

The software is designed using the traditional waterfall method while adopting evolutionary prototype technology. Waterfall method is selected because the requirement is well identified. There are many user experiences available about the requirements and expected outcomes. In that context, the requirement of the software is quite clear.

The evolutionary development methodology is also involved, as a project, it will never complete in a single trial. There will be many changes and improvements. Based on the time restrictions, all such improvements will be accommodated as much as possible to improve the quality of the software.

All the users except the tertiary user can log into the system through their logging passwords. The individual reservoir managers, who are in level 2, can carry out the limited operations only. The level 1 users will be the organization Heads and other top level officers who involved at higher level of Mahaweli Water Management Activities. All the users have to operate the system through the same interfaces, but with different privileges. Admin can perform any activity including new user registration and update of the user database.

3.1.2 Functional Requirements

Functional requirements are one of the most important part of the project as far as the end-user is concerned. In fact, immediate expectation is to get the expected output through the functionalities

of the system. Functional requirement is the answer to the question “what the system should do”. Functional requirements can be easily captured from use case diagrams, sequence diagrams, state charts, etc. Function consists of inputs, processing and the outputs relevant to the users’ expectation. Functional requirements are basically identified under modules which are described below.

- ✚ “Reservoir-Season Management” Module - This represents fixed reservoir parameters as well as the parameters that can be varied according to the selected season which depend on the start date of the cultivation season and the weather condition of the months which fall over the season, those again a function of the start date. The Mahaweli diversions, inflows from other sources are included in this module as user inputs. The variation of historical rainfall, the variation of cultivation extent and water-use efficiencies are also included as user inputs to optimize the solution. Other data for this module will be selected from the database.
- ✚ “Crop-Season Management” Module – This module is designed to select the data relevant to the crop that is going to be cultivated in the selected season. Apart from the crop data which are fixed for the selected crop, there are data which depends on the farmers practice. These practices are identical to the crop and the particular season. Mainly it is relevant to the paddy field preparation activities. For some type of paddy varieties, farmers use water for land soaking and land tillage. These parameters are included in this module, as this module handle the parameters relevant to the durations of the crop growth. Data for this module is partly from the database and partly user inputs
- ✚ “Database update” Module – The system should facilitate to update the database through the relevant interfaces. The system user has given the facility to either add new data, delete or edit data based on the privileges. Data can be added for the rainfall, reservoir, specific-yield, crop variety, evaporation, reservoir-season, crop-season, etc.
- ✚ “Presentation” Module – This is one of the most important and attractive part of the system. It takes the data from reservoir-season module and crop-season module. Then it processes the calculations for each month of the season and present the results on monthly basis. The results will be in tabular form as well as graphical form. This module makes the effectiveness of the whole system.

3.1.2.1 System Use case diagram

Use case diagram is the simplest and one of the most understandable UML diagram which explains the dynamic behavior of the system. It shows the actors and their relationship with use cases within the system. Use case is particularly useful in identifying the functionalities together with different users. It is a good communication tool to bridge the gap between designers and non-IT system users. *Figure 3.1* shows the high-level use case diagram of the proposed system.

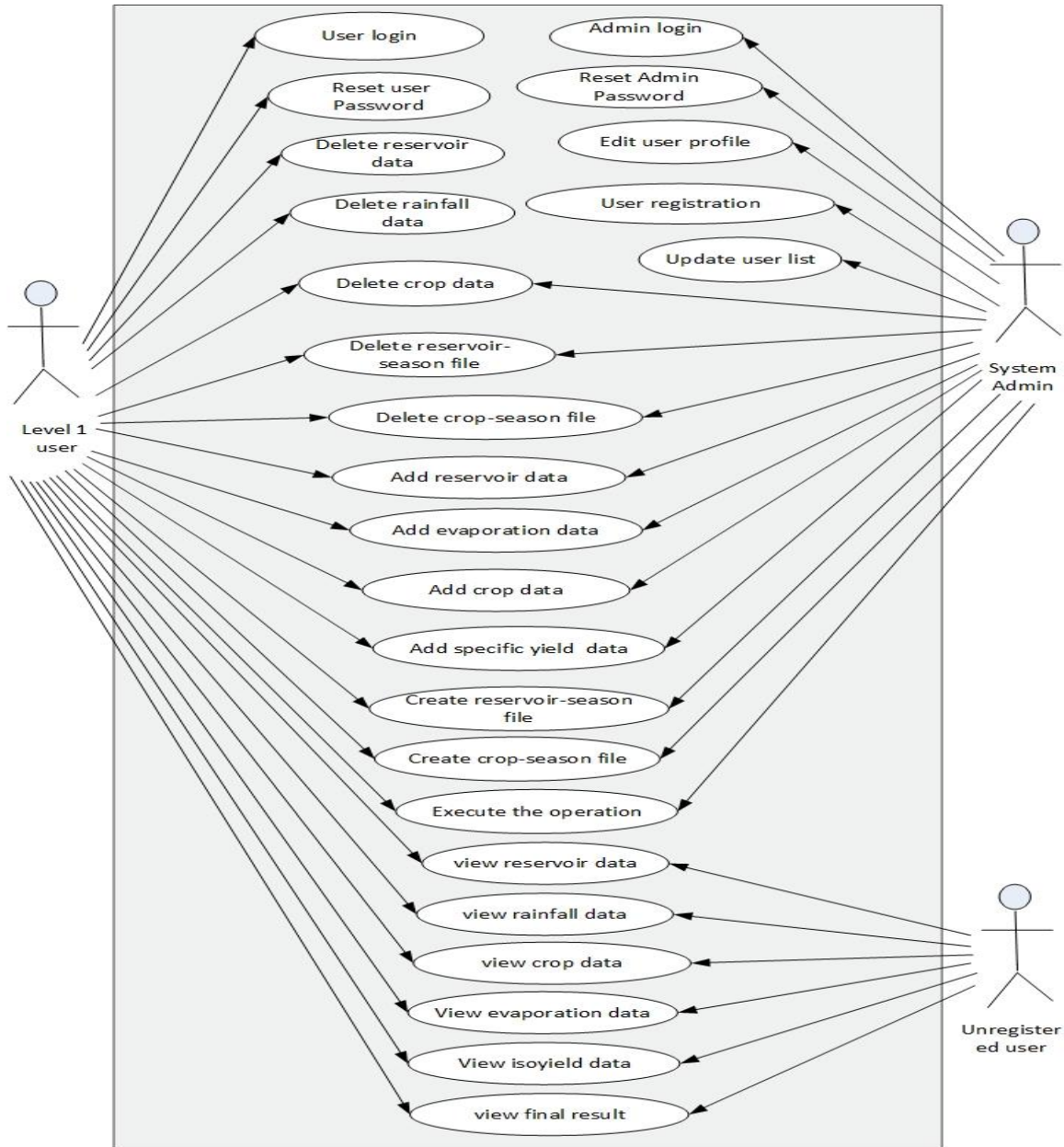


Figure 3-1 System high level use case diagram

3.1.3 Non-Functional Requirements

It is quite easy to describe the importance of the non-functional requirements by highlighting its impact over the system when non-functional requirements are not properly addressed. Nonfunctional requirements refer to a whole slew of attributes including performance levels, security, and the various parameters such as usability, reliability, and availability. Invariably, requirements definers get wrapped up in how the product is expected to function and overlook the non-functional requirements up to certain extent.

When such factors are not addressed adequately, seemingly proper product/system functioning in fact fails to function. For example, a system could be identified by the end users in such a slow, insecure, and difficult to use one that can cause mistakes which make the output unreliable and ultimately lead to abandonment.

That's the best way in which nonfunctional requirements do impact on system success.

According to the above description, it is the combine effect of the performance, maintainability, robustness, portability as well as the cost to achieve the above qualities. With the above factors in mind the non-functional requirements of the proposed system can be identified as;

- ✚ Performance – The system is expected to produce the results quickly. This requirement is achieved in such a way that the results can be obtained with a single interface of data selection. As the user, does not need to enter the data rather only to select two files from the database, system is at high performance level.
- ✚ Reliability – Data for the final output is received from the database. However, at the start, the user either should input the data to the database through the interfaces or should select already stored data. Various data validation techniques have been used when inserting the data through interfaces to the database. This will make sure the reliability of the results.
- ✚ Friendliness - This factor is highly achieved by making the system less complex. The modules for different purposes have been well identified and the relevant interfaces have been developed for each data input. Input interfaces are not complicated and not linked from one to another. At any time, user can complete one interface and leave the system and later can start from another interface.
- ✚ Portability - in relation to software, is a measure of how easily an application can be transferred from one computer environment to another. A computer software application is considered portable to a new environment if the effort required to adapt it to the new environment is within reasonable limits. With that context, reservoir management

programme is highly portable. Coding is done in PHP and MySQLi which are platform independent and not specific languages for hardware vendor. So it is very easy to change the environment with almost no cost.

- ✚ Effectiveness - The software should be effective in such a way that it should attract many users from the manual system. In fact, software should strong enough to generate new users as well. According to the design, it is quite confident that the software will be ranked as very effective by the users of all categories.

3.2 DESIGN

In the design stage of the software, one of the key aspect is to select a suitable architecture for the software. This will define the structured solution that meet all the technical & operational requirements, while optimizing attributes such as performance, security, and manageability.

This involves a series of decisions based on a wide range of factors, and each of these decisions can have considerable impact on the quality, performance, maintainability, and overall success of the application.

In finalizing the architecture, the functionality, usability, reuse, technological constraints and the tradeoffs are considered. Further the future trends are also considered.

By considering all those factors, Client/Server Architectural style was selected as the best. However, the Object-Oriented Architectural style and the Component based Architectural style will also be incorporated to decompose the design further into independent components that expose the appropriate communication interfaces. Using the object-oriented design approach for these components will improve reuse, testability, and flexibility.

3.2.1 Client/Server Architecture

The client/server architectural style describes distributed systems that involve a separate client and server system, and a connecting network. The typical client/server system involves a server application that is accessed directly by multiple clients, referred to as a 2-Tier architectural style.

Client/Server architecture indicated a graphical desktop UI application that communicated with a database server containing much of the business logic in the form of stored procedures, or with a dedicated file server. More generally, however, the client/server architectural style describes the relationship between a client and one or more servers, where the client initiates one or more requests, most probably using a GUI in this case, waits for replies, and processes the replies on

receipt. The server typically authorizes the user and then carries out the processing required to generate the result. The server may send responses using a range of protocols and data formats to communicate information to the client.

In the proposed system, it runs on a local machine, which intern consider as a local server. *Figure 3-2* below illustrates the system environment.

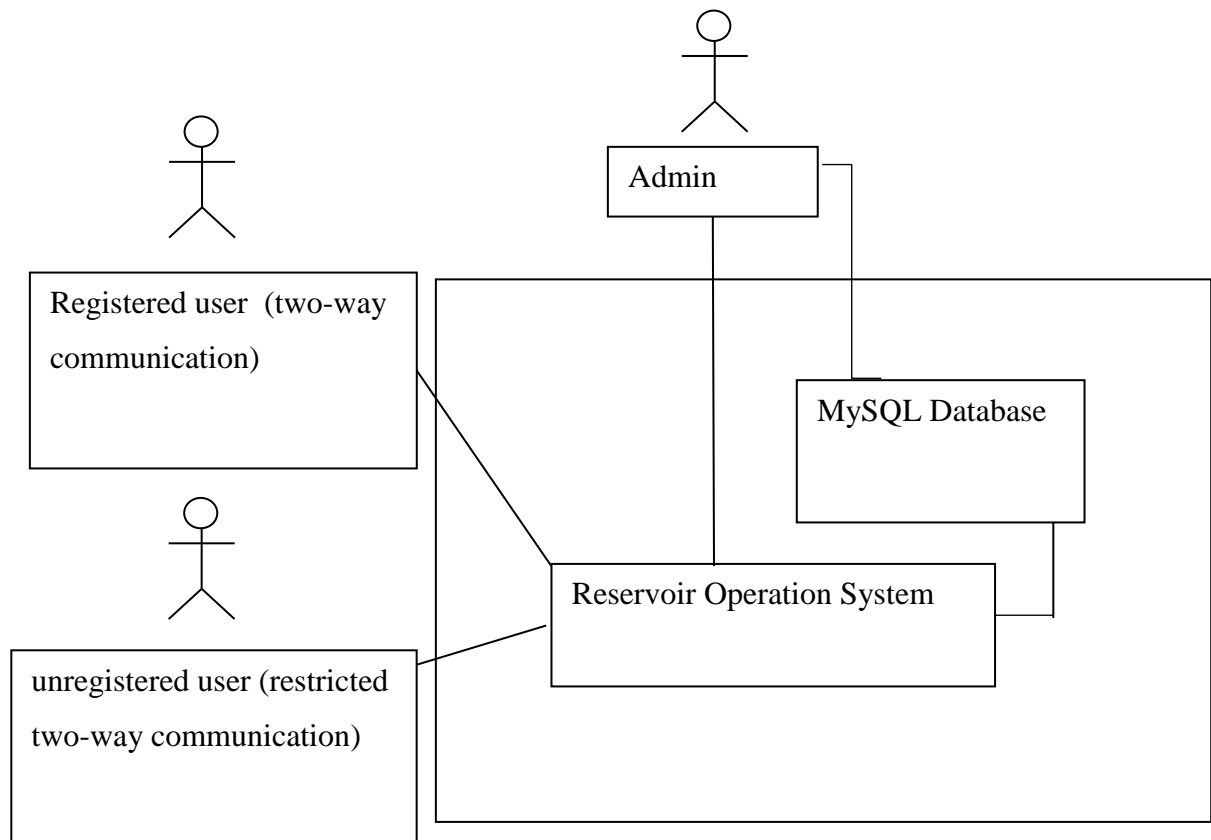


Figure 3-2 System Environment

There are key benefits of the client/server architecture over other systems.

- ✚ **Higher security.** All data is stored on the server, which generally offers a greater control of security than client machines.
- ✚ **Centralized data access.** Because data is stored only on the server, access and updates to the data are far easier to administer than in other architectural styles.
- ✚ **Ease of maintenance.** Roles and responsibilities of a computing system are distributed among several servers that are known to each other through a network. This ensures that a client remains unaware and unaffected by a server repair, upgrade, or relocation.

However, the traditional 2-Tier client/server architectural style has numerous disadvantages, mainly the tendency for application data and business logic to be closely combined on the server, which can negatively impact system extensibility and scalability, and its dependence on a central server, which can negatively impact system reliability. These disadvantages are not that rigorous for the system proposed for this study, as it is a closed one itself.

3.2.2 Object-Oriented Architectural Style

Object-oriented architecture is a design paradigm based on the division of responsibilities for an application or system into individual self-sufficient objects, each containing the data and the behavior relevant to the object. An object-oriented design views a system as a series of cooperating objects, instead of a set of routines or procedural instructions. Objects are discrete, independent, and loosely coupled; they communicate through interfaces, by calling methods or accessing properties in other objects, and by sending and receiving messages. The key principles of the object-oriented architectural style are listed below.

- ✚ Abstraction - Abstraction allows to reduce a complex operation into a generalization that retains the base characteristics of the operation. For example, an abstract interface can be a well-known definition that supports data access operations using simple methods such as Get and Update. Another form of abstraction could be metadata used to provide a mapping between two formats that hold structured data.
- ✚ Composition - Objects can be assembled from other objects, and can choose to hide these internal objects from other classes or expose them as simple interfaces.
- ✚ Inheritance - Objects can inherit from other objects, and use functionality in the base object or override it to implement new behavior. Moreover, inheritance makes maintenance and updates easier, as changes to the base object are propagated automatically to the inheriting objects.
- ✚ Encapsulation - Objects expose functionality only through methods, properties, and events, and hide the internal details such as state and variables from other objects. This makes it easier to update or replace objects, as long as their interfaces are compatible, without affecting other objects and code.
- ✚ Polymorphism - This allows to override the behavior of a base type that supports operations in the application by implementing new types that are interchangeable with the existing object.

In the design stage of the software, another key aspect is to select a suitable architecture for the software. This will define the structured solution that meet all the technical & operational requirements, while optimizing attributes such as performance, security, and manageability.

This involves a series of decisions based on a wide range of factors, and each of these decisions can have considerable impact on the quality, performance, maintainability, and overall success of the application. In finalizing the architecture, the functionality, usability, reuse, technological constraints and the tradeoffs are considered. Further the future trends are also considered.

By considering all those factors, Client/Server Architectural style was selected as the best. However, the Object-Oriented Architectural style and the Component based Architectural style will also be incorporated to decompose the design further into independent components that expose the appropriate communication interfaces. Use of the object-oriented design approach for these components will improve reuse, testability, and flexibility.

3.2.3 Design Methodology

Waterfall method was selected as the design methodology of the proposed reservoir operation system. Author, being member of the end-user group, is well aware of the requirements and there will be no much deviations of the requirements. It should be noted here that at the end of each phase, it is possible to review the work done and get back for the improvements.

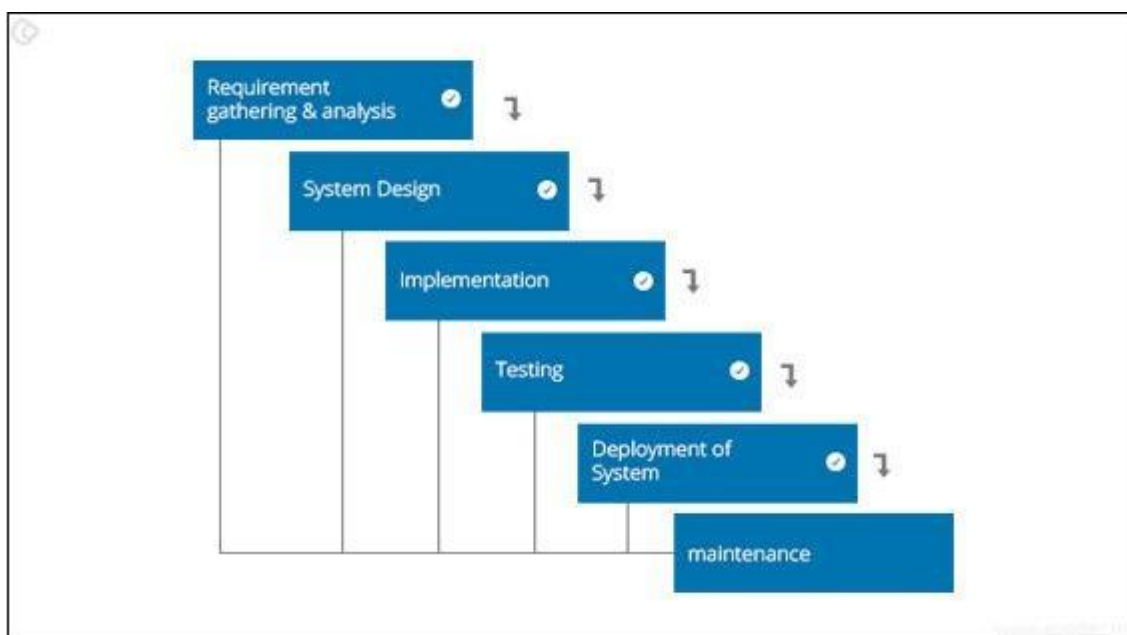


Figure 3-3 Illustration of the adopted development methodology

One key aspect to remember is that the phases will not overlap in the waterfall method. Paper prototype can be developed for the interfaces and flow of the system can be explained for the users. At this stage end-users were able to improve the interfaces if necessary by adding or removing the parameters.

To map the business flow and the database structure, UML diagrams, Class diagram and ER diagrams have been used. With the above diagrams, it is very convenient to work with the waterfall method, as the diagrammatic representations can help the people to understand the business flow. Exact method applied for the development is shown in the *figure 3-3* above.

3.2.4 Class diagram of the system

Class diagram represents the static view of the system and is the main building block of object-oriented modelling. Class diagram describes the system by showing the different classes, attributes of the classes, operations that the class could perform and the relationships between the classes. In *figure 3-4*, the class diagram of the system is shown with attributes and relationships.

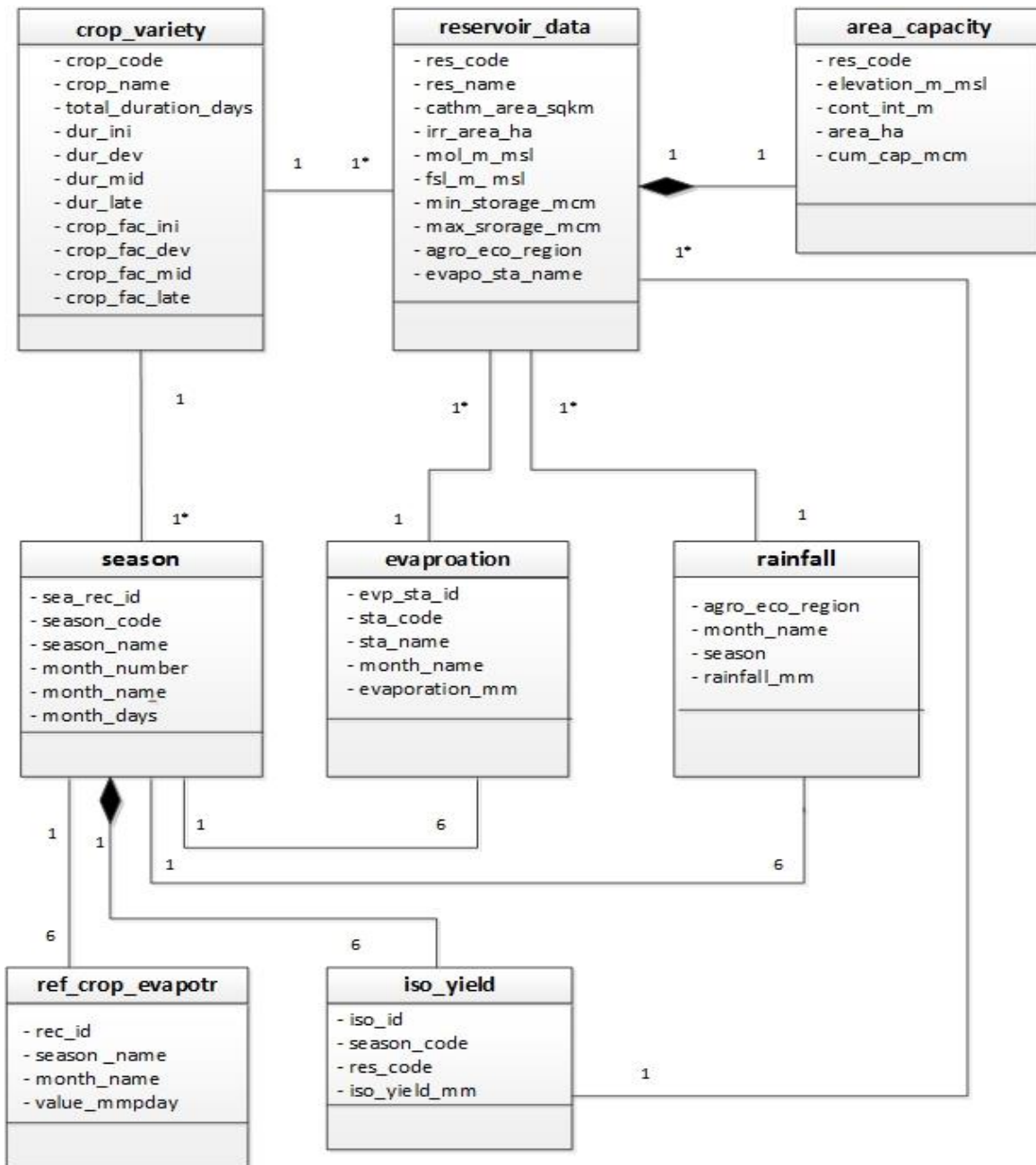


Figure 3-4 Class diagram of the system

3.2.5

3.2.6 Sequence diagrams

Sequence diagram for the reservoir operation system is quite simple as there is only user-system involvement. *Figure 3-5* and *Figure 3-6* show the sequence diagram for operation execution and the sequence diagram for adding reservoir-season data respectively.

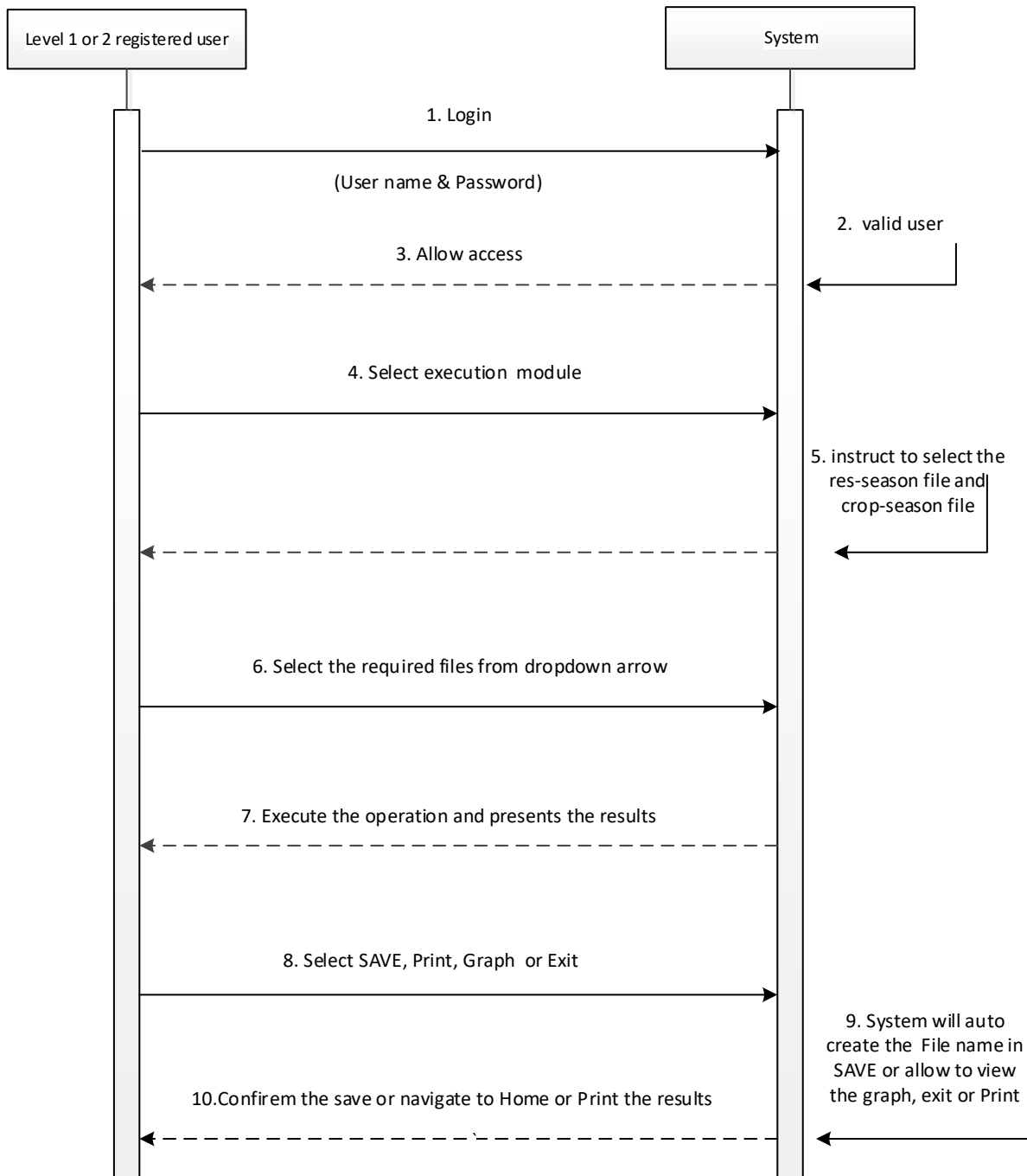


Figure 3-5 sequence diagram for execution of operation

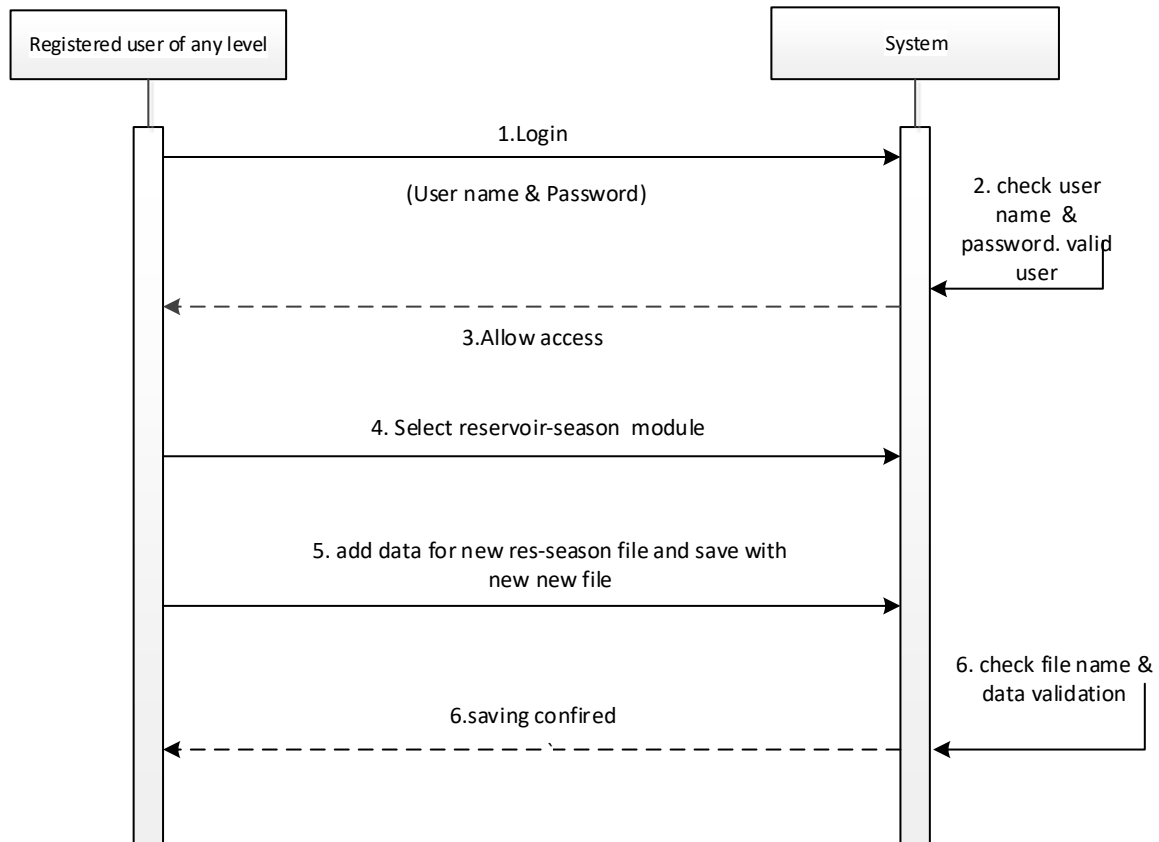


Figure 3-6 sequence diagram for saving the reservoir-season file

3.2.7 ER diagram

ER diagram shows most of the details of attributes of the classes and their interactions with the inheritance. ER diagram can be used to model and design of the relational databases, in terms of logic and business rules. The well designed Er diagram very much helpful in designing the database to match the functional requirements.

Cardinalities can be clearly identified from the ER diagram. By studying the ER diagram, the logical structure of the database or the connectivity of the data, can be visualized clearly. In the Reservoir operation system, the attribute *area capacity* is a composite attribute.

The conceptual ER diagram for the system is shown in *the figure 3-7*.

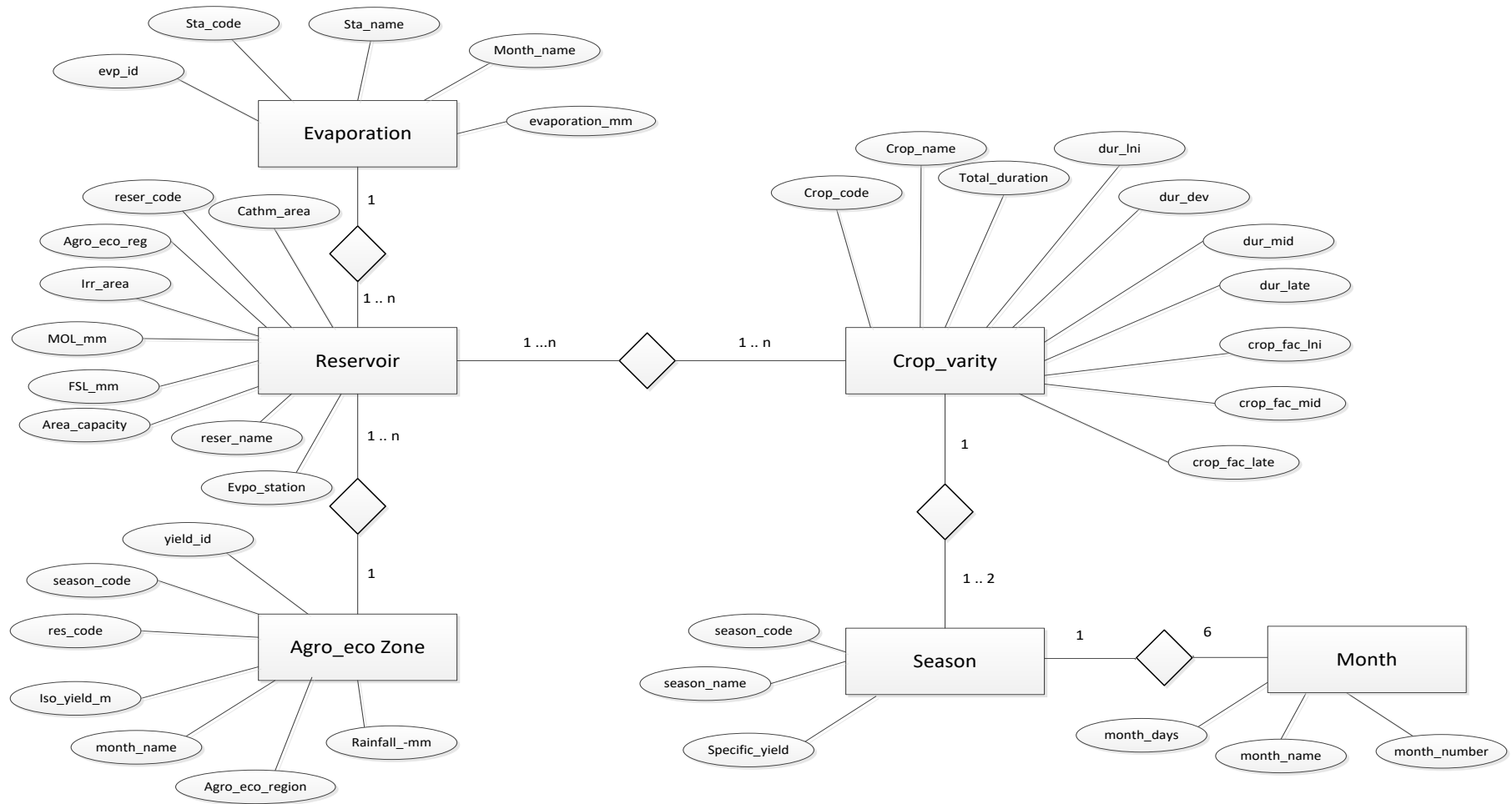


Figure 3-7 ER diagram for the reservoir operation system

3.2.8 User Interfaces

User interfaces have been developed by using the HTML with CSS. Starting from the Logging interface all other interfaces were designed to have a clear guidance to the user and return without any difficulty. Interfaces were designed to be in par with the sequence of the operations. Colors were selected to comply with the domain of the project.

Starting from the welcome page, user can proceed through the logging interface to the remaining steps of the system. *Figure 3-8* shows the welcome page and *figure 3-9* shows the logging interface.

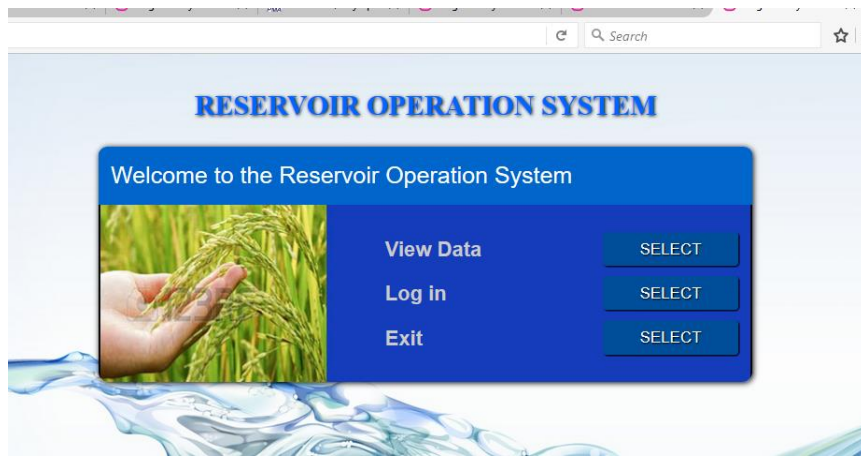


Figure 3-8 Welcome page of the system

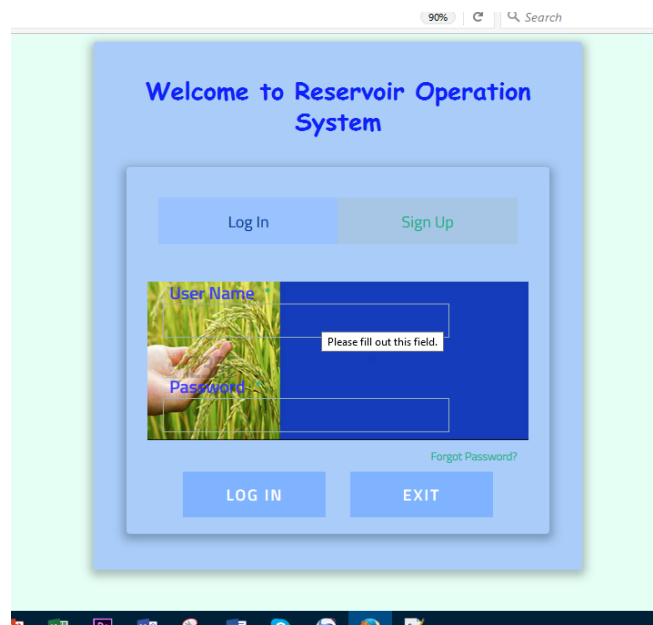


Figure 3-9 Logging interface

After the successful logging, a registered user can go through the system for data entry, edit, delete and calculation steps using the respective interfaces. *Figure3-10* shows the Home page when admin is logged and *figure 3-11* shows view data page. Admin is only the level 1 user who can perform all activities including the user registration in the system.



Figure 3-10 main menu after admin logging



Figure 3-11 view data page

As there are three user groups except the admin, once they logged into the system there will be different interfaces. *Figure 3-12*, *figure 3-13* and *figure 3-14* show the main interface once logged as level 2 user, i.e. organizational heads, level 3 user, i.e. Directors and level 4 user, i.e. Engineers of the Irrigation Department respectively.

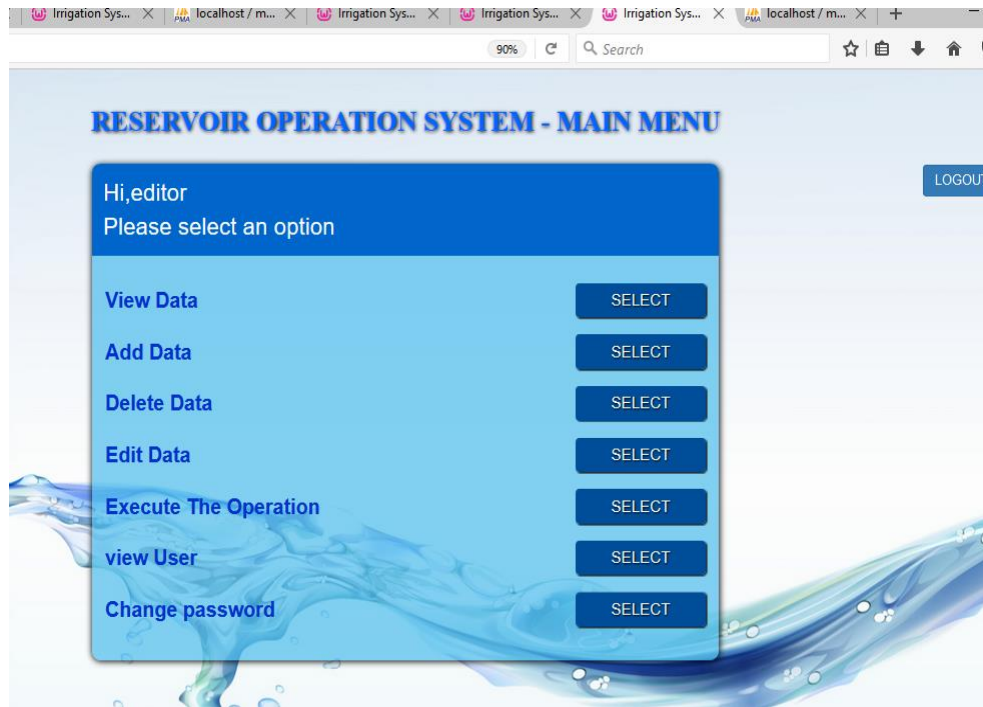


Figure 3-12 Main interface once logged as level 2 user

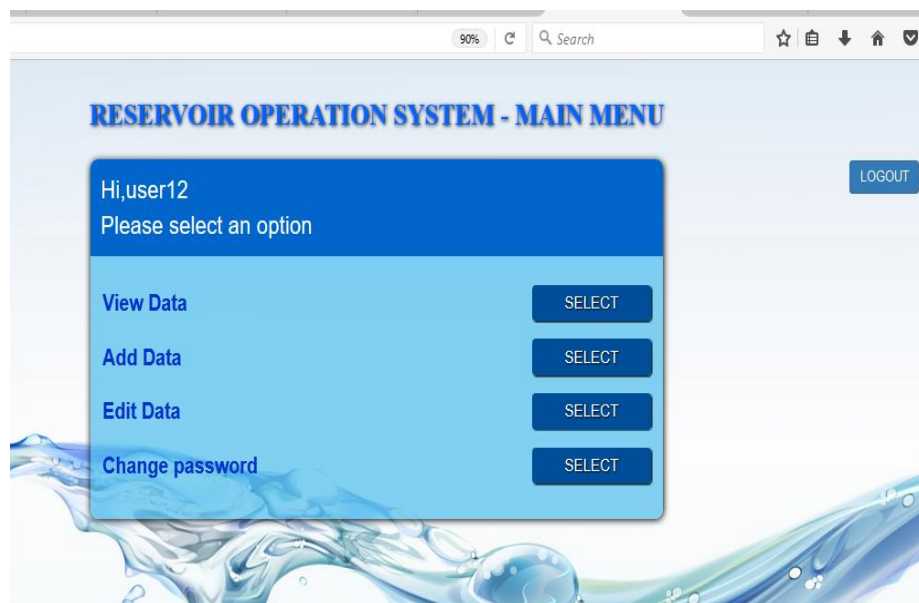


Figure 3-13 Main interface once logged as level 3 user

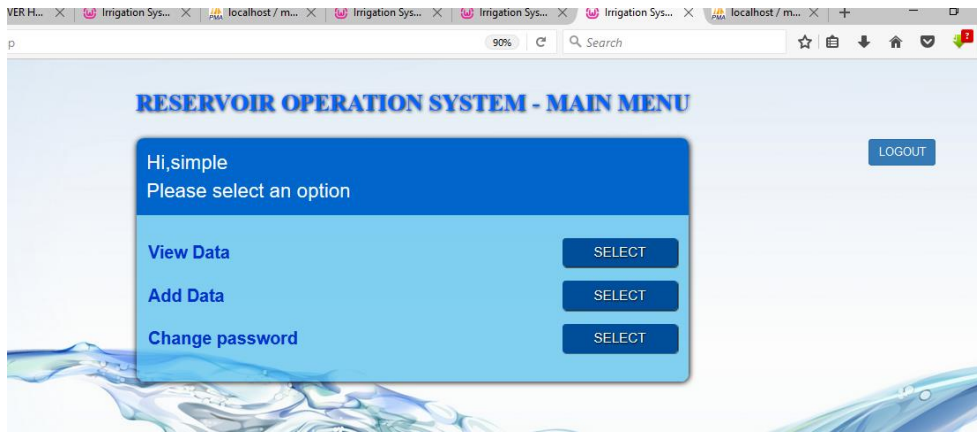


Figure 3-14 Main interface once logged as level 4 user

Through the above interfaces the registered users can go through the system and enjoy the functionalities based on their privileges. All the users can change their password as they were registered by the admin.

Only the admin can add new user or remove an existing user. However, the level 1 user can see who are the registered users.

There are two most important interfaces as far as the final result is considered. To get the final output, it is necessary to create two files, in reservoir-season module and crop-season module.

Those two interfaces are shown in figure 3-15 and figure 3-16.

Figure 3-15 Interface for adding reservoir-season file

Once a user save those two files, then user can execute the operation study to get the final results. This is presented in a table for six months of the season. The variation of most important parameters is displayed in a graphical form as well.

Figure 3-16 Interface for adding crop-season file

Figure 3-17 below shows the final results of the programme.

Month	Storage at the beginning (MCM)	Water level (m.MSL)	Catchment Inflow (MCM)	Mahaweli Inflow (MCM)	Inflow from upper Reservoir (MCM)	Irrigation Demand (MCM)	Losses (MCM)	Spilling (MCM)	Final Storage (MCM)	Field water Req. (mm)	Eff. Rainf (mm)	Status
APR	136.9	93.87	1.814	6	0	2.609	3.348	0	118.756	66.479	47.068	SUCCESS
MAY	118.756	72.83	0.725	11	0	52.704	3.776	0	55.002	400.923	8.777	SUCCESS
JUN	55.002	45	1.814	11	0	35.737	0.373	0	31.706	312.964	47.068	SUCCESS
JUL	31.706	67.95	0	4	0	42.063	1.812	0	12.14	312.966	0	FAIL
AUG	12.14	87.78	1.814	2.5	0	30.868	0.418	0	12.14	276.739	47.068	FAIL
SEP	12.14	87.78	0.363	0	0	3.658	0.395	0	12.14	27.214	0	FAIL

Figure 3-17 Final results of the programme

3.2.9 Database

Database of the reservoir operation system is designed to use with the MySQLi database features. There are eight tables to store the row data that will be obtained from the publications. These eight tables and their relationships can easily be mapped with the Class diagram and the ER diagram.

There are two more additional tables to store the data retrieved from the above eight tables through interfaces and by saving with file names. One file is to save the reservoir-season parameters and other file is to save crop-season parameters.

The final calculation is done using the data from these two tables. There is a flexibility to use any two combination of the files selected from these two tables.

The details of the tables are provided in APPENDIX B. *Figure 3-20*, below shows the database record where data from interface of *reservoir-season* is saved.

The screenshot shows the phpMyAdmin interface displaying a table with the following columns: save_id, file_name, res_name, cathm_area, aen, fsl, cfsf, iso_yield_mm, sb, season_d, season_n, evp, mol, mins, lah, srv, erv, cal, mon1, mon2, mon3, and m. The table contains 21 rows of data representing different reservoir-season records.

save_id	file_name	res_name	cathm_area	aen	fsl	cfsf	iso_yield_mm	sb	season_d	season_n	evp	mol	mins	lah	srv	erv	cal	mon1	mon2	mon3	m	
40	ajith	minneriya	238.000	DL1	93.700	137.700	240.034	70	10/11/2016	MAHA	kantalai	85.900	12.140	10080	558.8	15	642.62	oct	nov	dec	ja	
41	file 2	giritala	24.100	DL1	93.000	27.100	204.037	80	10/27/2016	MAHA	kantalai	82.000	2.400	3225	558.8	20	120	oct	nov	dec	ja	
42	file 3	kaudulla	371.000	DL1	73.170	128.280	38.405	60	08/15/2016	YALA	kantalai	67.000	12.050	5645	457.2	58	158	apr	may	jun	ju	
43	file 4	kantala	482.000	DL2	59.280	140.620	288.040	50	10/26/2016	MAHA	kantalai	82.500	28.310	7513	812.8	30	130	oct	nov	dec	ja	
44	file 5	minneriya	238.000	DL1	93.700	137.700	240.034	20	10/04/2016	MAHA	kantalai	85.900	12.140	10080	558.8	23	687.324	oct	nov	dec	ja	
45	file 6	minneriya	238.000	DL1	93.700	137.700	5	10	10/12/2016	MAHA	kantalai	85.900	12.140	10080	558.8	11	620.2679989999999	oct	nov	dec	ja	
46	file 7	giritala	24.100	DL1	93.000	27.100	5	10	10/12/2016	MAHA	kantalai	82.000	2.400	3225	558.8	3	575.564	oct	nov	dec	ja	
47	file 8	giritala	24.100	DL1	93.000	27.100	5	10	05/09/2016	YALA	kantalai	82.000	2.480	3225	457.2	3	575.564	apr	may	jun	ju	
48	file 9	kantala	482.000	DL2	59.280	140.620	5	10	05/12/2016	YALA	kantalai	82.500	28.310	7513	101.6	3	575.564	apr	may	jun	ju	
49	file 10	giritala	24.100	DL1	93.000	27.100	60	50	10/13/2016	MAHA	kantalai	82.000	2.400	3225	558.8	20	670.56	oct	nov	dec	ja	
50	asdw	giritala	24.100	DL1	93.000	27.100			11/08/2016	MAHA	kantalai	82.000	2.480	3225	127		oct	nov	dec	ja		
51	ajithax	minneriya	238.000	DL1	93.700	137.700			11/12/2016	MAHA	kantalai	85.900	12.140	10080	127	10	139.7	oct	nov	dec	ja	
52	ajith_2	minneriya	238.000	DL1	93.700	137.700			11/06/2016	MAHA	kantalai	85.900	12.140	10080	127		oct	nov	dec	ja		
53	Ajith_3	minneriya	238.000	DL1	93.700	137.700	135	25	11/08/2016	MAHA	kantalai	85.900	12.140	10080	127	12	142.24	oct	nov	dec	ja	
54	xabc	minneriya	238.000	DL1	93.700	137.700			11/06/2016	MAHA	kantalai	85.900	12.140	10080	127		oct	nov	dec	ja		
55	sdef	kaudulla	371.000	DL1	73.170	128.280			11/06/2016	MAHA	kantalai	67.000	12.050	5645	127		oct	nov	dec	ja		
56	ddos	giritala	24.100	DL1	93.000	27.100			11/06/2016	MAHA	kantalai	82.000	2.480	3225	127		oct	nov	dec	ja		
57	gdon	minneriya	238.000	DL1	93.700	137.700			11/06/2016	MAHA	kantalai	85.900	12.140	10080	558.8		oct	nov	dec	ja		
58	abdge	giritala	24.100	DL1	93.000	27.100	240.034		11/07/2016	MAHA	kantalai	82.000	2.480	3225	558.8		oct	nov	dec	ja		
59	abdgee	minneriya	238.000	DL1	93.700	137.700	240.034		11/07/2016	MAHA	kantalai	85.900	12.140	10080	558.8	10	614.68	oct	nov	dec	ja	
60	ASEDWR	minneriya	238.000	DL1	93.700	137.700			07/03/2016	YALA	kantalai	85.900	12.140	10080	457.2		APR	MAY	JUN	JUL		
61	today	giritala	24.100	DL1	93.000	27.100			25	11/08/2016	MAHA	kantalai	82.000	2.480	3225	558.8	10	614.68	OCT	NOV	DEC	JAN

Figure 3-18 Database record of saving the data through reservoir-season interface

The interfaces discussed above are only the key interfaces through which the user can go through the system. More details of the user interfaces and their functionalities are discussed in Appendix A under user manual.

4 IMPLEMENTATION

Implementation chapter explains the technical details behind the project and how it works. Reservoir operation system is a combination of climate conditions, reservoir properties and the crop being cultivated. System is expected to be used by the professionals in the reservoir management. In this chapter, sample codes and important flowcharts will be used for the explanation of the calculations. The IT literacy, Domain knowledge, adoptability and the user-friendliness of the coding language are considered.

4.1 TECHNICAL BACKGROUND

Behavior of three key sub domains will be explained here. As explained above, those are reservoir system, climate and the crop. Climate parameters are absolutely independent. One way crop and reservoir have independent parameters. However, when the crop combine with the climate, the parameters of the crop adjusts to suit with the climate. The reservoir is also an independent system. However, it should be noted here that depending on the cultivation season of the crop, reservoir should serve for the needs of the crop. Depending on the cultivation starting dates, reservoir will get different rainfall and evaporation rates which are important in reservoir operation. Finally, the system ends up with a harmonizing among climate, crop and reservoir. Above explained behavior of the reservoir and crop with the climate are greatly evaluated and included in the software system. Extremely useful information that the Author feel not fair if not presented, are discussed below.

4.1.1 Terms and Definitions

The key parameters involved in this programme are described below.

Catchment area – It is an extent or the area of land where all surface water from rain converges to a single point at a lower elevation. For the context of this study, catchment area is the area within which the rainfall received or any surface inflow received from upstream reservoir will contribute to the reservoir and not to any other reservoir. The lowest elevation of the catchment in case of the reservoir operation is at the reservoir sluice sill.

Elevation-Area-Capacity diagram (table) – This is a very important combination of natural property of any reservoir. It simply indicates the volume of the reservoir and the water spread area of the reservoir at any given height. This is the most critical natural quality of the reservoir. The water surface area affects the evaporation of water from the reservoir. On the other hand, storage gives an indication about the amount of water that is available in the reservoir.

Reservoirs are good if the rate of increase of storage is higher than that of the area. *Elevation-Area-Storage* relationship is determined by topographical surveys. Typical area capacity diagram for Giritale reservoir is shown in *figure 4-1* below.

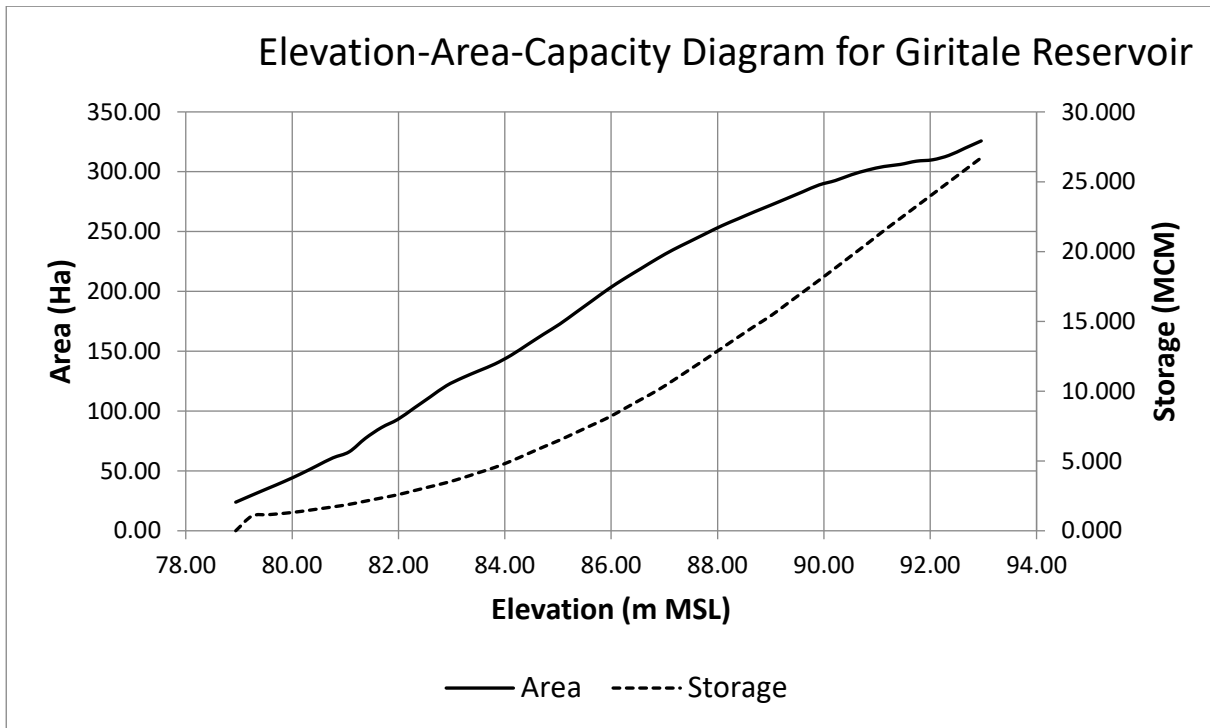


Figure 4-1 Elevation-Area_Capacity diagram for Giritale reservoir

Other terms and definitions used in the thesis can be visualized using the figure 4-2 below. Some key terms are defined next to the figure.

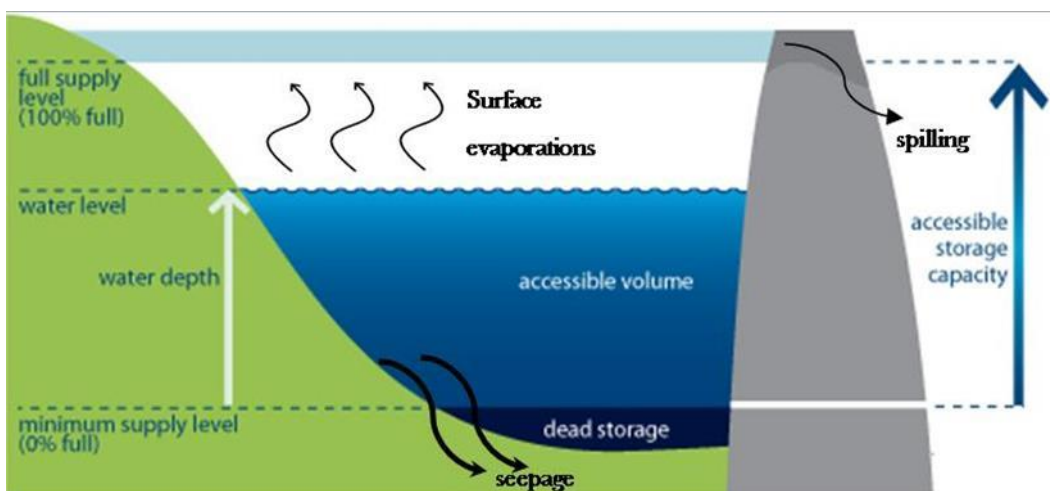


Figure 4-2 Typical cross section of a reservoir showing important parameters

Full Supply Level (FSL) – This is the maximum elevation up to which the water can be stored in the reservoir for the purpose of irrigation. Storage at FSL is called as *capacity*. If the water storage exceeds this level it will overflow as spilling.

Minimum Operating Level (MOL) - This is the minimum elevation beyond which the stored water cannot be used for the purpose of irrigation. Storage below the MOL is called as dead storage and can be used for the recreational purpose and environmental purposes only.

Command Area – It is the full area that is available for the cultivation under the reservoir. In some seasons, the cultivated area could be different from the command area due to water shortage in some seasons. In this study, a factor is incorporated to accommodate this reduction.

Application Efficiency – This term covers the loss of water that is not really used for the crop, though it is applied to the cropped field. One way of such loss is the deep percolation of water beyond the root zone of the plant.

Rainfall – Rainfall is not a perfectly predictable natural phenomenon. However, when the behavior of rainfall is studied over a long period, a statistical pattern could be discerned. Accordingly, it is possible to predict the amount of rainfall during any month up to a certain extent. In simple terms 80% probable rainfall for a given month is defined as, if for a selected period of ten years, if a particular month received a rainfall of given value in eight years, then the probability of receiving that rainfall is 80%. Longer the duration of data availability higher the accuracy. For the operation study, irrigation Department recommends to use the 75% probable rainfall. These rainfall values are available for different regions of the country, called as *Agro-ecological regions*, published by the Agrarian Services Department. *Figure 4-3* shows the Agro-ecological regions of Sri Lanka on which the reservoirs under study is superimposed.

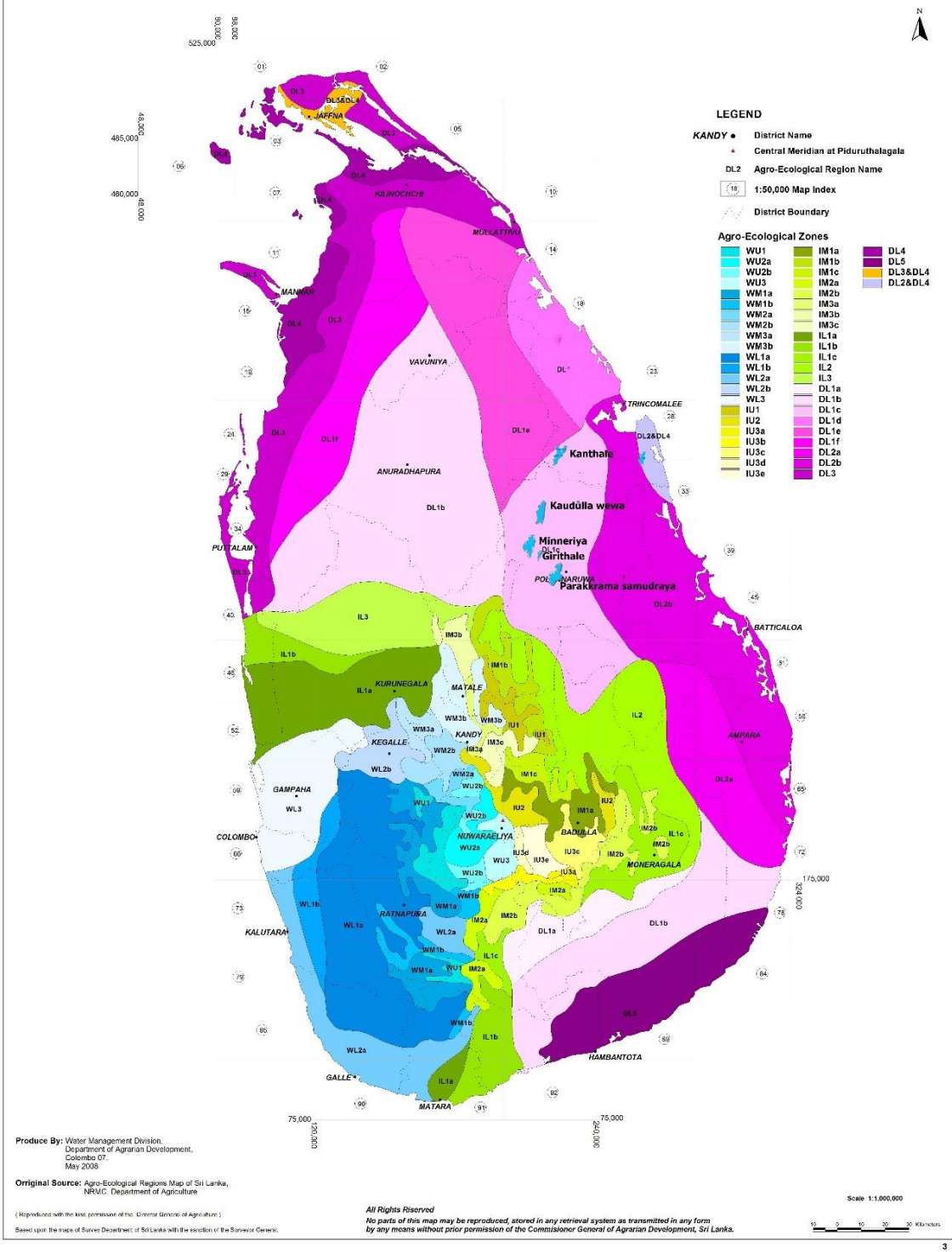


Figure 4-3 Selected reservoirs on the Agro-ecological regional map of Sri Lanka

Specific Yield – Specific yield is that part of rainfall per unit area of the catchment, assumed to be contributed to the storage of reservoir through runoff. The specific yield contours, called as *iso-yield contours*, for both MAHA and YALA seasons have been developed for the country

by the Agrarian Services Department. It is also noted here that there are upper and lower limits for the ratio of seasonal yield to the seasonal rainfall as the yield is developed from longer data set and the rainfall may not be the same data set. The recommended values for these upper & lower limits are 35% and 8% respectively. If this is elaborated further, if the *specific yield* value is 700ham per sq.km and if the rainfall is 1400ham per sq.km, then the ratio is 50%. However, the upper limit is 35%. Hence the maximum catchment inflow due to rainfall for that season should be 35% of 1400. This limitation is due to the wide range of interpolations done in developing the iso-yield maps, wide coverage of rain gauges, uncounted water losses etc. However, these conservative limits ensure the availability of water for the cultivation with minimum risk. The standard unit of the specific yield is hectare-meter per square kilometer (ha.m/sq.km).

Conveyance Efficiency – This term covers the loss of water, during the conveyance through the canal system from reservoir to the paddy fields.

Evapotranspiration - The transpiration through the crop together with the evaporation of water from the soil where the crop is planted is called as evapotranspiration. The evapotranspiration rate is normally expressed in millimeters (mm) per unit time. The rate expresses the amount of water lost from a cropped surface, which includes the contribution from both the crop and soil, in units of water depth. The time unit can be an hour, day, decade, month or even an entire growing period or year.

As one hectare has a surface of 10000 m² and 1 mm is equal to 0.001 m, a loss of 1 mm of water corresponds to a loss of 10 m³ of water per hectare. In other words, 1 mm day⁻¹ is equivalent to 10 m³ ha⁻¹ day⁻¹

Evapotranspiration is calculated using a *reference crop* according to the FAO standards.

The *reference crop evapotranspiration* [ET_{ref}] is defined as “**The evapotranspiration from an extended surface of an 8 to 15 cm tall green grass cover of uniform height, actively growing, completely shading the ground and not short of water**” by the FAO.

4.1.2 Water requirement for the Cultivation

In this study, calculation of the water requirement for the cultivation consists of three main components. They are;

- ✚ Actual water requirement of the crop. This is called as Crop Water Requirement (CWR).
- ✚ Water necessary for the land preparation. This component depends on the practices of the farmers on each crop. Land preparation water requirement is divided into land soaking water requirement and land tillage water requirement.

- ✚ Water received from the rainfall. All the above water requirements cannot be fulfilled from the rainfall. In fact, only part of the rainfall is effective for the plants.
- ✚ The difference between the above-mentioned water requirements and the effective rainfall shall be delivered from the reservoirs. The Net requirement will be increased based on the system efficiencies

4.1.3 Crop Water Requirement

Crop water requirement plays the major part of the software being developed. It is complex and will be explained below.

CWR for the crop being cultivated is a combined effect of the climatic factors and crop characteristics. Climatic factors are temperature, wind-speed, humidity and solar radiation. Crop characteristics are mainly the leaf-area-of the crop at any given time and its growth stage. Crop characteristics are combinedly addressed through the crop factor, K_c . The effect of the climate and the behavior of the crop are clearly explained by the text book *Guidelines for computing crop water requirements*, by FAO.

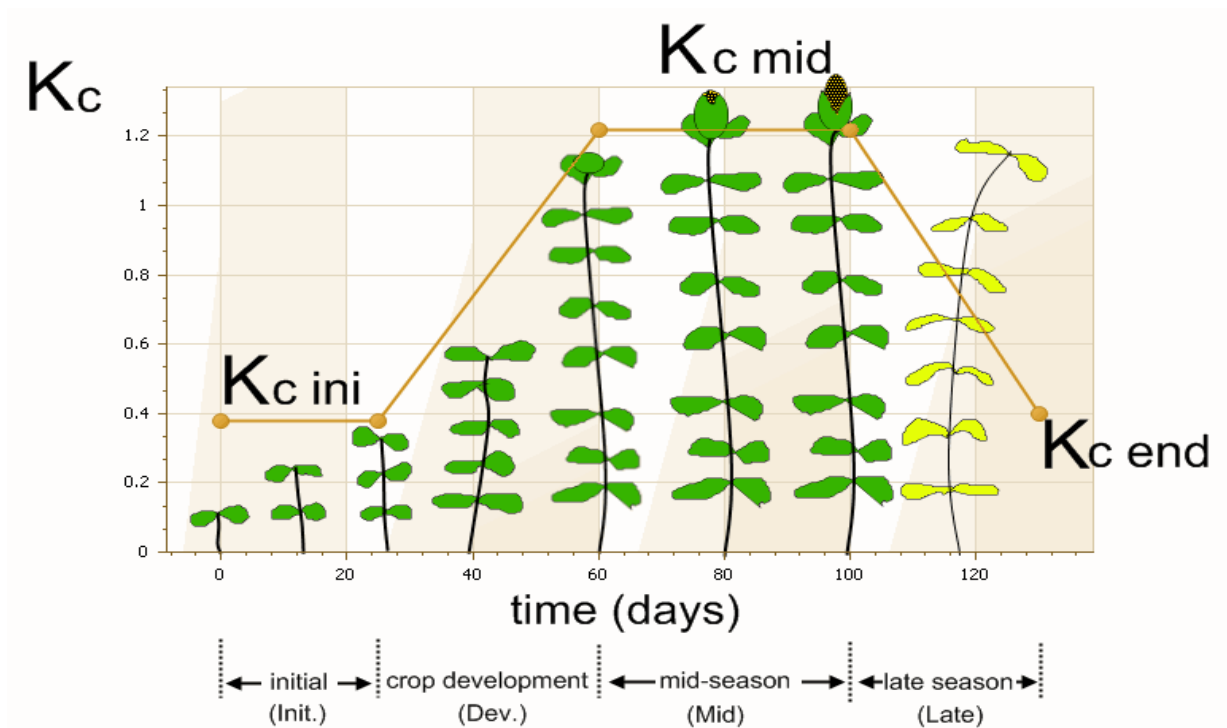


Figure 4-4 variation of the crop factor with growth stage

Figure 4-4, above illustrates the variation of the crop factor of a seasonal crop according to its growth stage.

Paddy has well identified four growth stages. Water consumption of the paddy depends on these growth stages as well as the climatic factors. For each growth stage, water consumption of the crop is linked through crop-factor. This water demand is a result of the combined effect of

evaporation from the cropped surface, i.e. soil, and the transpiration from the crop itself. At the early stages evaporation part is higher as the majority of the soil surface is exposed to the sun. However, when the plant grows up, most of the area is covered by the leaves of the plant so reducing the evaporation part while increasing the transpiration part. These variations are clearly explained in the FAO book listed under the references.

There are no other alternative accepted methods available in calculating the crop water requirement. In the FAO calculations of CWR, climatic factors and the crop related factors are considered. In the FAO calculations, crop water requirement is represented as a function of potential evapotranspiration and the crop factor.

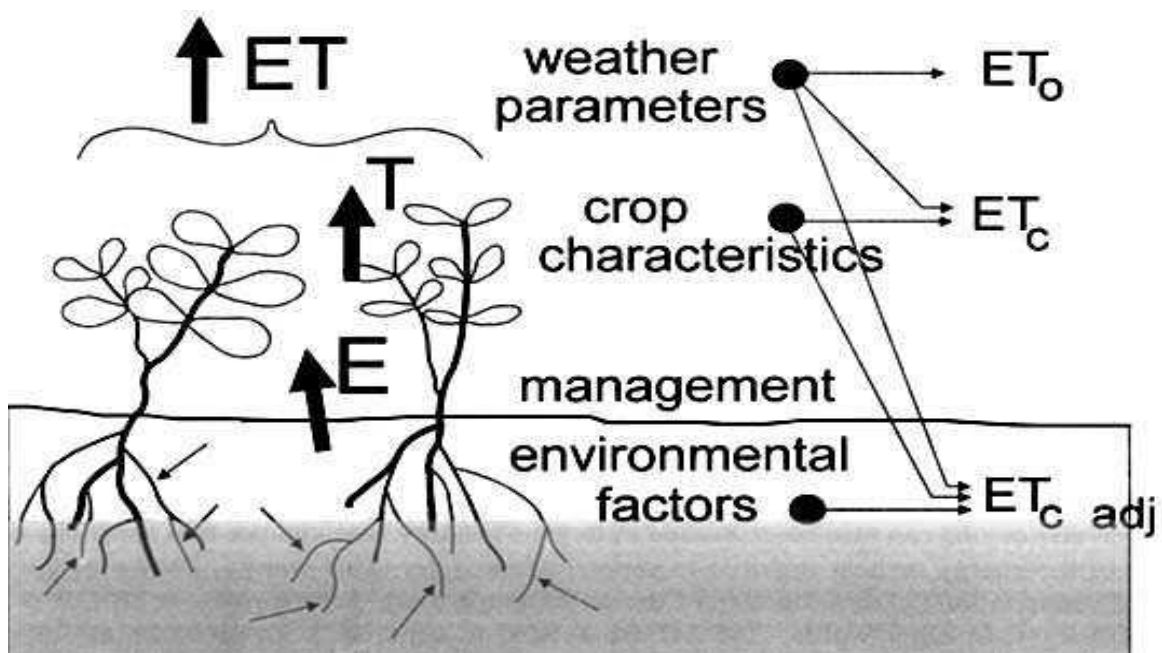


Figure 4-5 Factors affecting on the CWR

The water requirement for the crop, technically termed as CWR, can be calculated using the K_c and ET_0 values together with the human factors and environmental factors. The factors affected on CWR is illustrated in figure 4-5 above.

The value of the crop factor, K_c , depends only on the growth stage of the crop. For paddy, four growth stages have been identified. Crop factor and the duration vary for each growth stage for any variety of paddy. There are several factors that influence on this variation. Those factors are also shown in figure 4-5.

When the crop factor is known, it is possible to calculate potential evapotranspiration of the crop using the equation given below.

$$ET_{crop} = K_c \times ET_0$$

ET_0 values are published by the FAO for the month and K_c values can be found on growth stage basis. Duration of the crop growth stages which obviously do not run parallel with the calendar months. As the reservoir operation study goes on monthly basis, there will be more than single K_c value for any month.

It should be noted here that rainfall, evaporation and even reservoir operations are calculated based on calendar months. In almost all the cases growth stage either goes beyond a single month or part of the month. This makes the calculation very complex, as the climatic parameters are available on monthly basis while crop parameters do not. The latter totally depends on its growth stage. The *figure 4-6* shows the shifting of growth stages from the monthly calendar.

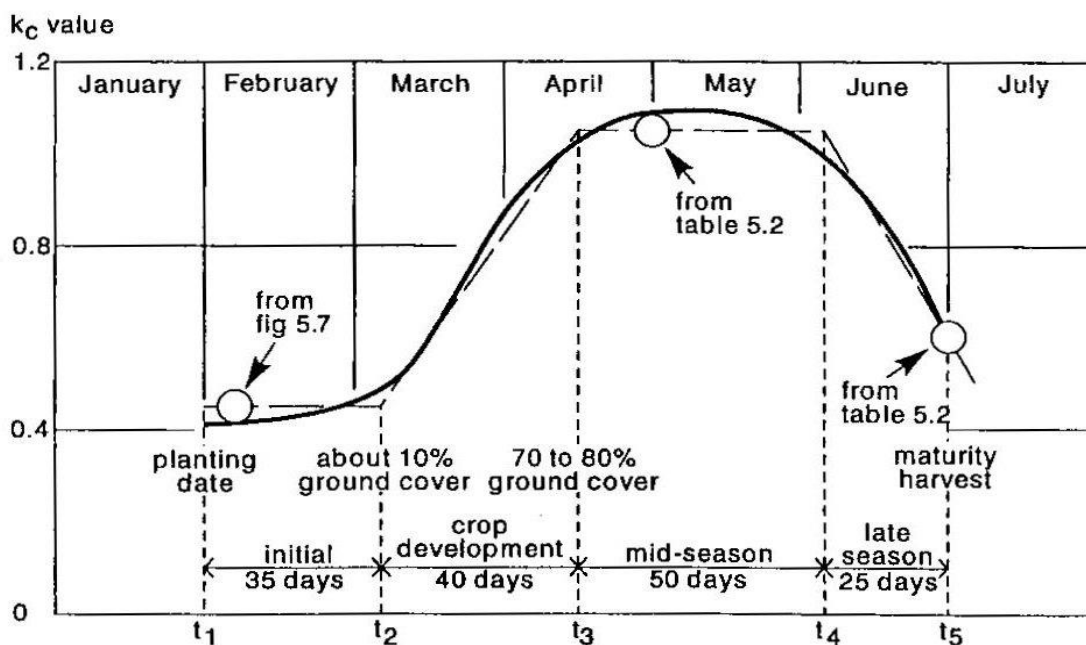


Figure 4-6 Illustration of the independent overlapping of the crop factor with months

4.1.4 Climatic and Seasonal influence

The effect of the climate and the starting date of the cultivation is another important area of the programme. All the climatic data are published on the calendar month basis. For the reservoir operations, monthly rainfall inflows, evaporation losses, etc. can be calculated from the climatic data. In some cases, climatic data are published on seasonal basis. One example is specific yield is published on seasonal basis. There are techniques to convert this seasonal values to monthly values. Depending on the start date of the cultivation, values to be selected will be changed.

4.1.5 User Flexibility

Reservoir is the remaining most influential factor in the programme. In many practical situations, if the management failed to select the right combination of crop and its starting date, the cultivation will fail. This failure is due to the rigidity of the reservoir parameters. In the proposed system, there are many steps have been taken to make the effect of rigid reservoir parameters more flexible. Some of such steps are listed below.

- ✚ Introduction of rainfall variation factor - By introducing this factor, the inflow to the reservoir can be adjusted as feel by the system user. This adjustment will ultimately change the water availability in the reservoir.
- ✚ Adjustments to the area of cultivation – The extent of the cultivable land is a fixed property of the reservoir. This is called as command area. However, in some cases full extent of the command area will not be cultivated in the selected season due to lack of water in the reservoir. In the proposed system, there is a provision to adjust the area cultivated as a factor of command area. This will facilitate the system operators to select a cultivable area suitable to the available storage of the reservoir.
- ✚ Different Artificial inflows – This is another provision that makes the proposed operation study programme more flexible. If the reservoir operators were unable to manage the cultivation with previous adjustments or if they are not happy to make previous changes, the reservoir operators can add artificial inflows on monthly basis.

Apart from the above adjustments, all other parameters are fixed properties of the reservoir. It is the responsibility of the reservoir operators to select a balance solution based on the crop, climate and reservoir parameters.

4.1.6 Calculation Flowcharts

Reservoir operation system calculates the crop water demand, evaporation, catchment inflow calculations based on the guidelines of *Guidelines for computing crop water requirements* [1], *Crop yield response to water* [2] and *Irrigation Headworks* [3]. As explained at the beginning of this chapter, some of the calculations are very complex. Calculation of crop water requirement is most complex one. Calculation of seasonal yield is not that difficult. Two flowcharts, describing how to calculate the monthly inflow to the reservoir due to rainfall and crop water requirement are included in this chapter for the illustration. *Figure 4-7* illustrates the calculation process of monthly water inflow to a reservoir due to rainfall. The next figure,

Figure 4-8, illustrates the calculation process of the end of month storage of a reservoir by considering all possible inflows and outflows.

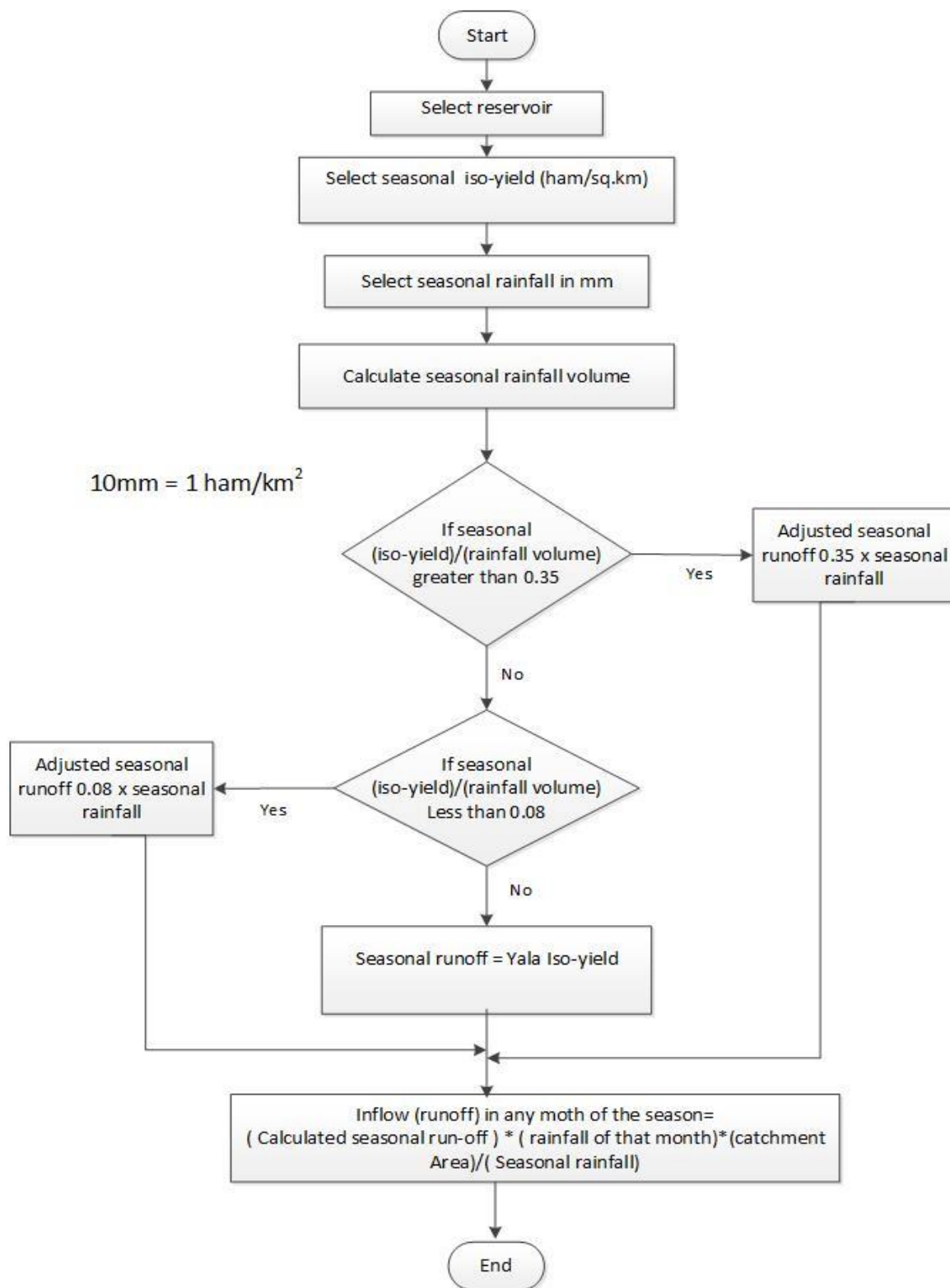


Figure 4-7 Flowchart for calculating the monthly inflow due to rainfall

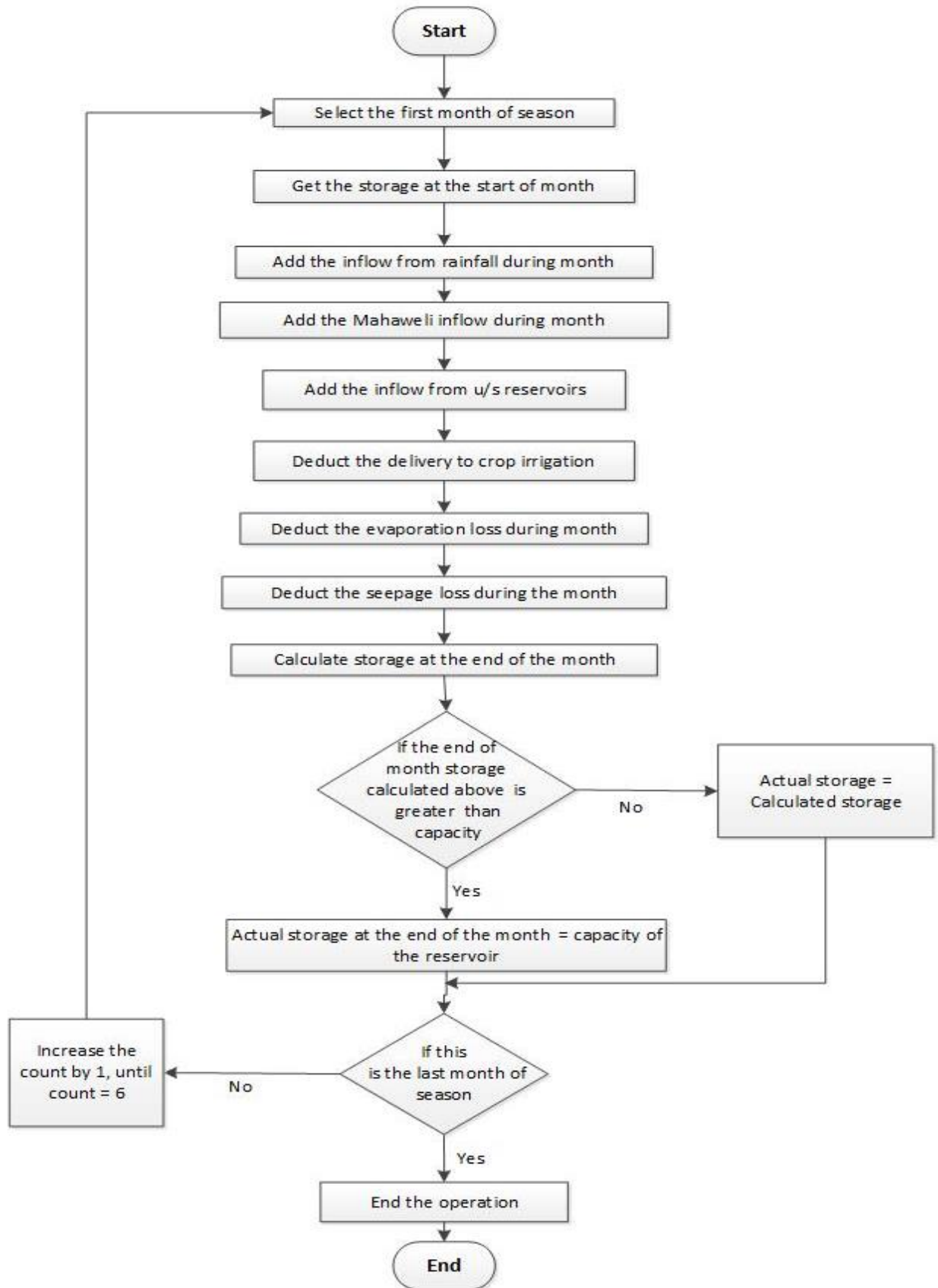


Figure 4-8 Calculation of the final storage of the reservoir

4.2 SOFTWARE IMPLEMENTATION

The programming language of the reservoir operation system is PHP. AJAX and JQUERY were used as supporting tools. Support of the MySQLi is used for the database. Interfaces and the pages have been developed using the HTML. Data captured through the interfaces will be stored in the database and retrieved for the calculations. The final output is again stored in the database.

4.2.1 Sample code

A single sample code is given in this chapter. This includes part of the password validation, character validation, empty field verification, etc. used in user registration.

```
var regx_chr = /^[a-zA-Z_-]*/; //not [a-zA-Z0-9_-]
if(uname.length<5 ||uname.length>10 || uname.search(regx_chr)==-1) {
    window.alert('The Name must contain between 3 and 10 characters
    \nAccepted only Letters and "-" & "_');
    uname.select();    // select the field for Name
    return false;
    exit();
}
var atposition=email.indexOf("@");
var dotposition=email.lastIndexOf(".");
if (atposition<1 || dotposition<atposition+2||
otposition+2>=email.length)
{
window.alert("Please enter a valid e-mail address \n
atpostion:"+atposition+"\n dotposition:"+dotposition);
return false;
}
if ((password.length < 5) || (password.length > 15))
{
window.alert("Your Password must be 5 to 15 Character");
return false;
}
```

5 EVALUATION AND TESTING

5.1 TESTING

Testing is the one of most important part of the product that will develop the confidence to put the product into the service. Any product whatever its facilities and attractiveness, if failed in the testing, it could not be delivered to the customer.

Software testing and other coding techniques [10] explains the objective of the testing, professional to be involved and the basic steps that should be followed in software testing. Finally, it specifies when to start testing and when to stop the same According to [10], testing is the process of evaluating a system or its component(s) with the intention to find whether it satisfies the specified requirements or not. In other words, testing critically evaluates the system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

Testing process includes verification and validation. Verification is done at all the stages of system development to check whether the system is developed correctly. Validation is done after implementation on the executable system to check the system requirements given by the clients at the initial stages against the features of the final system.

Testing if however, an endless process which eventually has to be stopped with a balancing condition of time, budget and acceptability by the user. It should be noted here that engineering the design process to make the product have less defects may be more effective than engineering the testing process. This chapter discusses the tests done for the reservoir operation system as per the standard procedures.

5.1.1 Functional Testing

Functional testing was done to ensure that the software is capable of giving the expected output. Four main outputs of the functional testing can be listed as;

- ✚ To ensure the functionality that the intended application is meant to perform.
- ✚ Creation of the test data according to the initial specifications.
- ✚ To check the output according to the test data.
- ✚ Comparison of the results against the expected outputs.

5.1.2 Unit Testing

The primary objective of the unit testing is to ensure that the individual code segments are working properly. No test data is essentially used at this stage. Unit testing has to be done for

all possible options, before merging the individual units. Normally this is a responsibility of the developers on their own codes.

5.1.3 Integration Testing

Integration testing is done for combined parts of the software to ensure their combination is working properly. The integration can be bottom-up integration or top-down integration.

5.1.4 System Testing

After integrating all modules and components of the system, the system will be checked for its quality standards as a whole under the system testing. As far as the industry is concerned, this part of testing has to be carried out by a specialized professional team.

Followings are the key benefits and highlights of the system testing.

- ✚ It is the first step of the life cycle of the development process, where the application is tested as a single unit.
- ✚ Application is tested as much as closer conditions to the its real deploying environment.
- ✚ Business requirements as well as the application architecture will be tested, verified and validate under the system testing.

5.1.5 Acceptance Testing

This is the formal testing mechanism with respect to user needs, requirements, and business processes conducted to determine whether or not the system satisfies the acceptance criteria. Further this enables the user or the authorized entity to determine whether or not to accept the system. There are many test level and criterions for the acceptance testing. However, finally the successful acceptance testing will enable the user to accept the product without hesitation. Acceptance Testing is performed after the system testing and before making the system available for actual use.

Final acceptance testing was further verified with the top-level management and the Engineers who have been involving in reservoir operations and decision making in water allocations. One of the key factor highlighted at the acceptance testing was the flexibility of the software to execute the operation study with various user input parameters. Capability to select crop-season data and reservoir-season data independently for the operation study is also highly appreciated. Further the provision to edit and save the reservoir- season data and the crop-season data while keeping the initial file unchanged is also very much useful for the decision makers in doing small changes and compare the results. Finally, the presentation facilities used to view the results in graphical form as well as a hard copy is highly appreciated by the end users. There were several parameters that should be adjusted to suit to the actual ground conditions of the

field. Those parameters were adjusted at the time of model calibration using the data available in the Irrigation Department. Later acceptance testing was done after validating the model with such adjustments. Finally, the acceptance testing was successful and the results and the information provided by the system was to the extreme satisfaction of the higher-level Engineers involved on the reservoir operations.

5.2 TEST PLAN AND TEST CASES

IEEE Standard for Software and System Test Documentation [11] specifies the test documentation, test plan, test case specification, test design specification, etc. in extremely systematic and a comprehensive manner. Even though, it is quite impossible to fully adhere to the standards in the individual project Author has taken measures to do the testing and its presentation to achieve the objectives that the IEEE standards expects. Attention have been drawn in planning the test cases and carrying out the testing to get the answers for the following standard fields.

- ✚ **Test case ID** - Unique ID for each test case.
- ✚ **Test priority (Low/Medium/High)**- Test priority for business rules and functional test cases are considered as medium or high.
- ✚ **Test case title** - This is the title assigned for the test case such as *verify login page with valid username and password*.
- ✚ **Pre-condition**: Any prerequisite that must be fulfilled before execution of this test case. For the successful execution of the test case all pre-conditions shall be met.
- ✚ **Expected result**: This field simply answer to the question of what *should be the system output after test execution?* This may include the expected result including message/error that should be displayed on screen.
- ✚ **Post-condition**: This field simply answer to the question of what *should be the state of the system after executing this test case?*
- ✚ **Actual result**: This field describes the system behavior after test execution.

5.2.1 Test case for failure to save the Reservoir-Season data without a Required field
Saving the Reservoir-season data is one crucial function as that will be selected for the calculations. It is very important to have all the controls over the data and functionalities of this module. Because of its importance, test was carried out to save the data without any filed. It is summarized in *table 5.1* below.

Test case ID	3	Test Priority	high		
Test case Title – Verify the data retrieved from tables “reservoir data”, “iso-yield”, “rainfall” and “season”, cannot be saved without filling all the required fields in the “save” table.					
Pre- Condition – User must be a registered user and logged to system. Data should be stored in tables “reservoir data”, “iso-yield”, “rainfall” and “season”.					
Testing details					
Step	Description of step	Test data	Expected result	Actual result	Status (pass/fail)
1	Navigate to reservoir-seasonal data interface	Fail to save the data using the reservoir-season interface			
2	Select a reservoir from drop down				
3	Select the start date of the season from calendar				
4	Input the all required fields except a single required field				
4	Click save button		<i>Please fill <data field name></i> message will appear	Expected message appeared.	PASS
Post-Condition - The success message will disappear in short while. Interface remains until select another option.					

Table 5-1 Test case for failure to save reservoir-season file without a single required field

5.2.2 Test case for Retrieving data and saving the Reservoir-Season file

Table 5-2 shows the test case for creating the reservoir-season file to be used for the operation study once the user logged with proper username and password. Test case for creating the crop-season file is similar to this except the differences in data inputs.

Test case ID	4	Test Priority	high		
Test case Title – Verify the correct data is retrieved from different tables and saved them under the given file name in a single table					
Pre- Condition – User must be a registered user and logged to system. Data should be stored in tables “reservoir data”, “iso-yield”, “rainfall” and “season”.					
Testing details					
Step	Description of step	Test data	Expected result	Actual result	Status (pass/fail)
1	Navigate to reservoir & seasonal data interface	Data to be saved using the reservoir-season interface			
2	Select a reservoir from drop down				
3	Select the start date of the season from calendar				
4	Input the filename and other required fields				
4	Click save button		<i>File “ “ saved successfully message will appear</i>	Expected message appeared.	PASS
Post-Condition - The success message will disappear in short while. Interface remains until select another option.					

Table 5-2 Test case for saving the reservoir-season data with valid data

5.2.3 Test case for valid user logging

Table 5-3 shows the test case for user logging with proper username and password. The next test case is to confirm the rejection of user logging with either incorrect username or incorrect password. This test case is illustrated in the APPENDIX C.

Test case ID -	1	Test Priority	high		
Test case Title – Verify registered user logging with correct username and password					
Pre- Condition – User must have valid username & password.					
Testing details					
Step	Description of step	Test data	Expected result	Actual result	Status (pass/fail)
1	Navigate to logging page	Userna me and passw ord			
2	Input the correct username				
3	Input correct password				
4	Click logging button		User should be able to log to the site	User is navigated with congratulation message	PASS
Post-Condition - User is validated with database and successfully login to the system					

Table 5-3 Test case for registered user logging.

As described in the above sub sections, test cases were done to check the functionalities of all the scenarios. Complete list of the test cases, done on scenario based testing is included in the APPENDIX C.

6 CONCLUSION AND FUTURE WORK

This chapter summarizes the Author's in-depth view about the project harmonizing all the factors. Especially the extent that the objective of the project being achieved, the difficulties encountered, limitations and maintenance and development issues in future. Author is fair enough to discuss the achievements, failures and difficulties encountered during the project work.

6.1 DIFFICULTIES ENCOUNTERED AND ACHIEVEMENTS

At the time of undertaking the project, Author has a fair knowledge about the variables of the domain and format of the final output. However, there was no clear idea about the programming language, interface requirements, testing procedures etc. After accepting the project proposal by the UCSC, Author went through the many websites to find out most convenient language for the project. As Author, did not have much knowledge about other programming languages PHP language was selected after discussing with the Project supervisor. Other reasons for selecting the PHP is its potential for future expansions of the project and easiness to couple with HTML and AJAX. It should be noted here that the project is associated with mathematical calculation and the security wise the project is less vulnerable. In that extent, one of the earliest and remarkable achievement is the selection of the correct language.

There was no great difficulties in requirement gathering, as Author practically involves with the reservoir operations for long period though there was no any kind of software application for this. However, Author discussed with the senior officers of the Water Management Divisions of Irrigation Department and the MASL to get the requirements with wider coverage. Next challenge is the identification of use cases, design the database and design the interfaces to suit the use cases. According to the requirements gathered, use cases were finalized and converted them into functional units through interfaces. However, the database design was changed for three times and a comprehensive database was finally developed. With a proper database and the understanding about the domain, it was not quite difficult to design the prototype of the interfaces.

One technique used for the development of the interfaces was, backward designing and check the interfaces for forward functionality. This method made the project little bit easy in interface design. As the Author is not involved professionally in software development, the interface inputs are checked using the Microsoft Excel for the expected output before starting the coding. One way this is a kind of testing prior to real coding.

With perfect identification of the interfaces, the next challenge is to do the coding. With the advice of the project supervisor, UCSC staff and after searching the web, it was not very much difficult to draft the coding. This drafting includes when and where the AJAX, and JQUERY to be included with the basic PHP codes and where to include PHP code segments inside the HTML codes.

After finalizing the above steps, it was not that difficult to do the coding for different modules. Then the individual modules were tested through the interface inputs, database updates and even with echo command. Finally, those modules were coupled in forward integration. Enthusiastically outputs came even with deviations. These deviations were subsequently checked, corrected and verified.

Last part is the testing. Even though it was not much theoretically difficult as the total project is handled by a single person with full understanding, time limitation is greatly affected for this activity. Testing was done for many test cases with different data inputs.

6.2 USEFULNESS

There is no doubt, the Reservoir Operation System will be a very handy tool for the reservoir managers and high-level officers of the Irrigation Department and Mahaweli Water Management Secretariat.

6.3 FUTURE WORK

Reservoir Operation System software was developed by targeting the existing domain characteristics of the Mahaweli Water Management system. However, there will be more suggestions in future mainly due to technical developments as well as social and organizational changes. Some of the potential future needs will be;

- ✚ Application for a crop of more than six months.
- ✚ Develop for a season deviated from the traditional cultivation season.

In addition to above it is much attractive to have a good logging system couple with a organizational website.

REFERENCES

- [1] R.G. Allen, et al., *Guidelines for computing crop water requirements*, Rome: Food and Agriculture Organization of the United Nations, 1980.
- [2] P Steduto, et al., *Crop yield response to water*, Rome: Food and Agriculture Organization of the United Nations, 2012
- [3] A.J.P.Ponrajah, *Irrigation Headworks*, Colombo: Irrigation Department, 1982.
- [4] P.Witharana et al., *Watersheds of Sri Lanka*, Colombo: Department of Agrarian Development Sri Lanka, 2011
- [5] CROPWAT 8.0 [Online]. Available:
http://www.fao.org/nr/water/infores_databases_cropwat.html. [Accessed: Sept. 12, 2016].
- [6] CLIMWAT [Online]. Available:
http://www.fao.org/nr/water/infores_databases_climwat.html . [Accessed: Sept. 12, 2016]
- [7] CRIWAR 3.0 [Online]. Available:
http://www.bos-water.nl/index_bestanden/Page568.html . [Accessed: Jun. 23, 2016]
- [8] M.Bos, et al., *Water Requirements for Irrigation and the Environment*, Wageningen: Springer Science & Business Media, 2008
- [9] CropWaterUse [Online]. Available:
<http://cropwateruse.net.au/default.aspx?ReturnUrl=%2f> . [Accessed: Sept. 16, 2016]
- [10] Software testing and other coding techniques [Online]. Available:
https://www.tutorialspoint.com/software_testing/software_testing_quick_guide.htm
- [11] IEEE Standard for Software and System Test Documentation [Online]. Available:
<http://standards.ieee.org/findstds/standard/829-2008.html>
- [12] Software testing standards [Online]. Available:
http://www.ufjf.br/eduardo_barrere/files/2011/06/SQETestCaseSpecificationTemplate.pdf

APPENDIX A USER MANUAL

A 1 Welcome page

After selecting the Reservoir Operating System icon from the website, the welcome page will appear. *Figure A-1* shows the welcome page of the reservoir operation system.

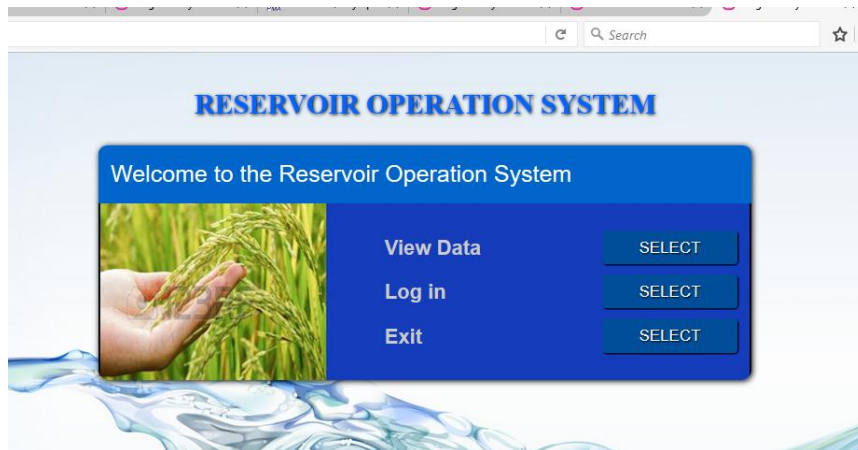


Figure A-1 Welcome page of Reservoir Operating System

A 2 Logging page

After appearing this page, user either can view the data through the interfaces, log in to the system with username and password or exit to the main website. In case if the user just wants to view the data, those functions will be explained later of this chapter. Consider a user who wants to log into the system. *Figure A-2* shows the logging window when a registered user opt to login to the system.

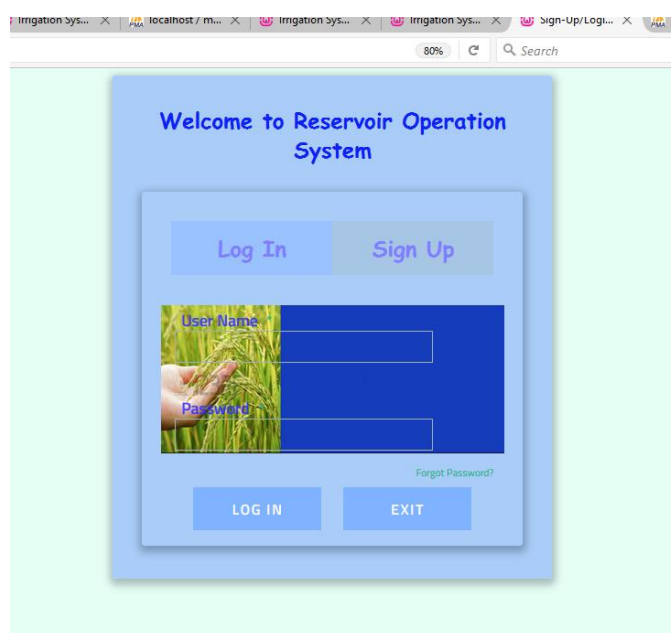


Figure A-2 user logging page

The user with correct username and password can log into the system. However, the unregistered user will be redirected to the same page. If the user wants to exit, the user will be directed to the welcome page.

A 3 user with successful logging

The successful user, after logging, will be directed to the main page of the Reservoir Operating System based on the user's privilege level. Screen shot of the Main Page for different user groups are given in the *figure A3 to figure A 6* below.



Figure A-3 Main page of the system for level 1 user or Admin

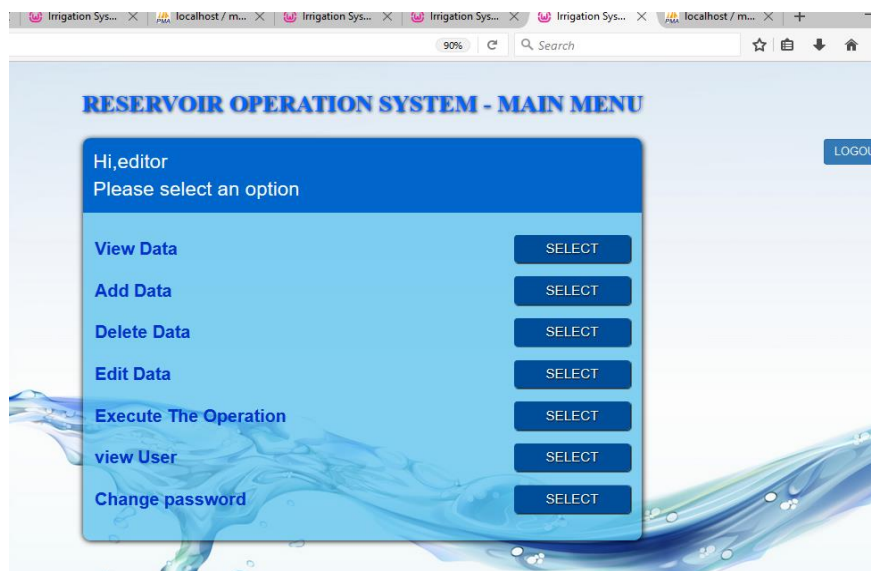


Figure A-4 Main page of the system for level 2 user

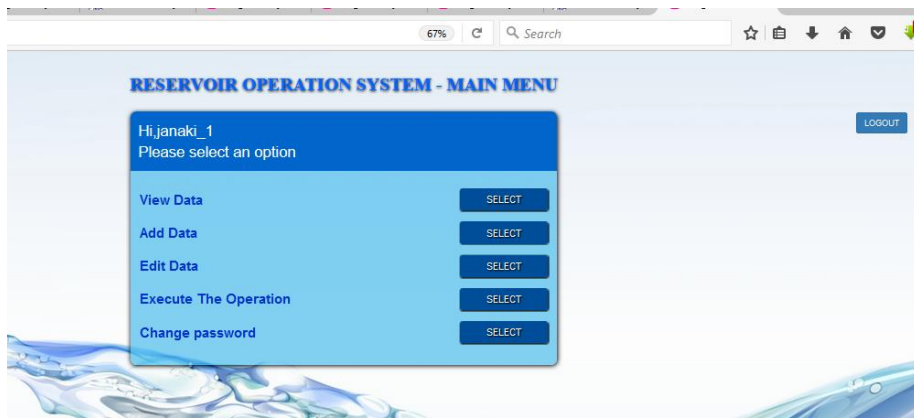


Figure A-5 Main page of the system for level 3 user

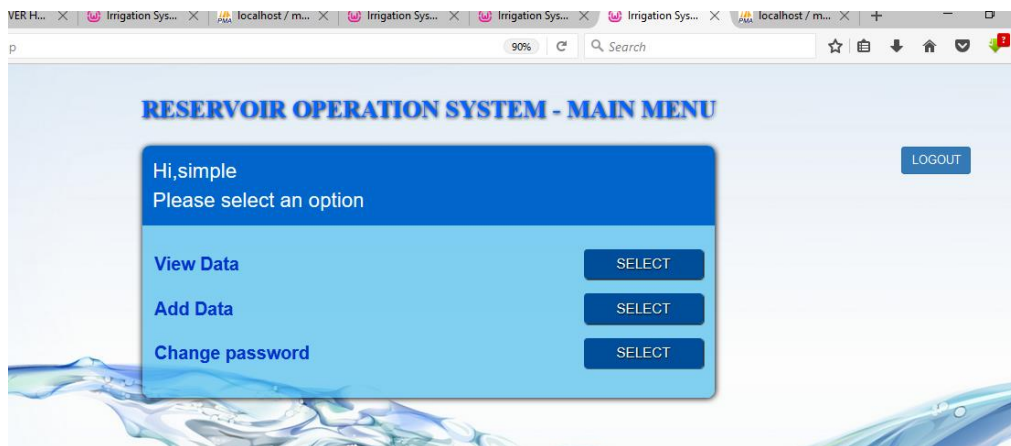


Figure A-6 Main page of the system for level 4 user

A 4 VIEW data

User with successful logging, can select either to view the data, Add the data or Execute the operations. First option is view the data. After viewing the data, the user will be more comfortable in deciding whether to add the data or not. There are eight interfaces through which the user can view the data. They are;

- ✚ View reservoir data.
- ✚ View rainfall data.
- ✚ View evaporation data.
- ✚ View specific yield data.
- ✚ View crop data.
- ✚ View reference crop evapotranspiration data.
- ✚ View reservoir-season data.
- ✚ View crop-season data.

The registered user or even the unregistered user can view all above information through respective interfaces. All the interfaces are very simple and almost same in its functionality. The user can view the data and go back to the main page when required. User has to select the base parameter on which the data to be viewed is pinned. More description is given with the respective interfaces which are given in the figures below.

Figure A 7 shows the main page which display all the available view data options for user at any level. This page will come once the user selects the “view data’ option of the main menu. The user can go back to the main menu once he selects the “return to min menu”.



Figure A-7 Interface facilitates the user to select any option to view all categories of data

After appearing this page, the user can select any option listed in the menu. When user select an option data relevant to that option will be displayed or the user will be instructed to select the particular main data field.

Figure A 8 below shows the interface to view the rainfall data. In the Reservoir Operation System, rainfall is tied on the “Agro- Ecology” field, so the user has to select the Agro-ecological area where he wants to see the rainfall data. Once it is selected, the data in the database will be displayed. At the same time the selection box will be changed to a light rose (hex #) color as illustrated in the figure A 8 below.

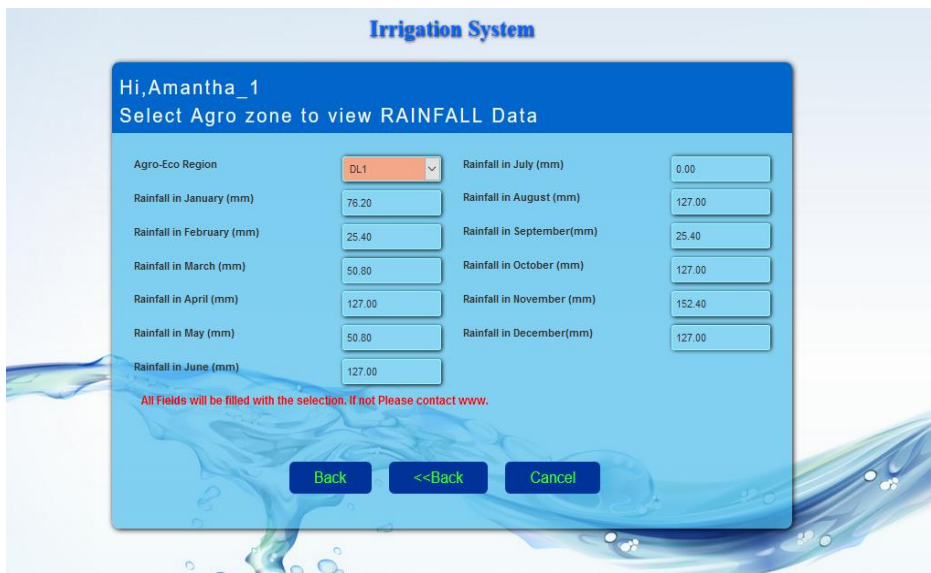


Figure A-8 interface to view the rainfall data

After viewing the data, the user either can go back two steps to the HOME page or a single step to the VIEW DATA page, figure A-7, to view some other data by clicking the appropriate buttons. Further if the user wants to view the same data fields by changing the selected option, still the user can do it by clicking the cancel button in the UI shown in the figure A -8 shown above. As described in the previous paragraph, other interfaces available to view the data are shown below. The functionality of all are almost similar, except in some interfaces the user has to select two parameters using two drop down menus. Figure A-9 is the interface relevant to view the crop data.



Figure A-9 interface for viewing the crop data

There are two interfaces which are slightly different from all other interfaces of view data. The data for these two interfaces are saved in the database from the retrieved data from several tables through the interfaces. Data of these two interfaces are directly used for the execution of the operation. Option field for these interfaces are file names under which the data is saved. The design of these interfaces are purposely made different from others to highlight the usefulness and to highlight the data for these views are not pure row data but partially includes the calculated information of the respective data. *Figure A -10* and *A -11* shows these interfaces.

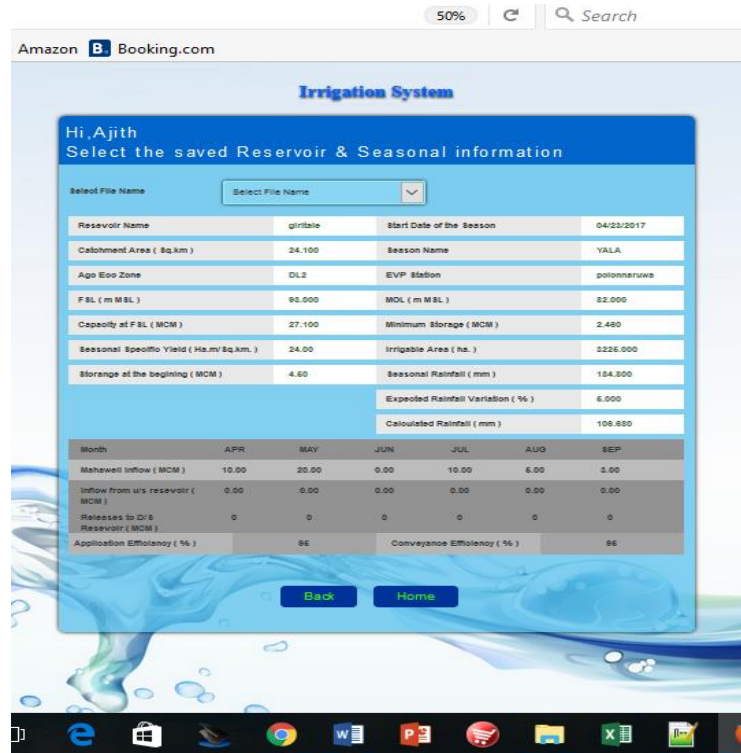


Figure A-10 interface for viewing the reservoir-season data



Figure A-11 interface for viewing the crop-season data

A 5 ADD DATA

User with successful logging, can select the option to Add the data. If the user feels that the data available in the database is not matched with his/her requirements, this facility will make the user more comfortable. There are seven interfaces through which the user can add the data. They are listed below. Add data is also common facility for all the users irrespective to the user level.

- ✚ Add reservoir data.
- ✚ Add rainfall data.
- ✚ Add evaporation data.
- ✚ Add specific yield data.
- ✚ Add crop data.
- ✚ Add reservoir-season data.
- ✚ Add crop-season data.

The registered user can add the data and go back to the main page if required. User has to select the base parameter on which the data to be added is tied on. Some of the interfaces are given in the figures below. In adding the data, user must be careful to fill all the required fields. Main page of add data, after successful logging is shown in *figure A -12* below.



Figure A -12 Main page to add data

The user can select any of the option from the main page of “add data” at a time. Once user selects an option he will be navigated to the next relevant page. *Figure A-13*, below, shows the interface for adding the rainfall data. When the data is saved the success message will appear in blue color, whereas the failure message will appear in red color.

One important point to keep in mind is that, there are tables in the database to which it is not possible to add new record before updating some other table. Due to this restriction, there are interfaces that cannot be really used to add the data, before adding key parameter of that interface through some other interface. For example, before adding the reservoir data, it is necessary to select the evaporation station and the agro-ecological region to which the reservoir belongs. If the data of those two fields were not added, user should add those data through “add evaporation data” and “add specific-yield data” interfaces before adding the reservoir data.



Figure A-13 Interface for adding the rainfall data

At this level, user can save the data, cancel or go back to the main page of add data. If the user does try to save the data without completing any single required field or filled with invalid data, the system will inform the user with alert message and highlighting the required fields or invalid fields. In many cases the message will highlight the valid data field, in case of error due to invalid data. The figure A-14 below shows the interface with system reaction when trying to save the data with invalid data or incomplete fields.

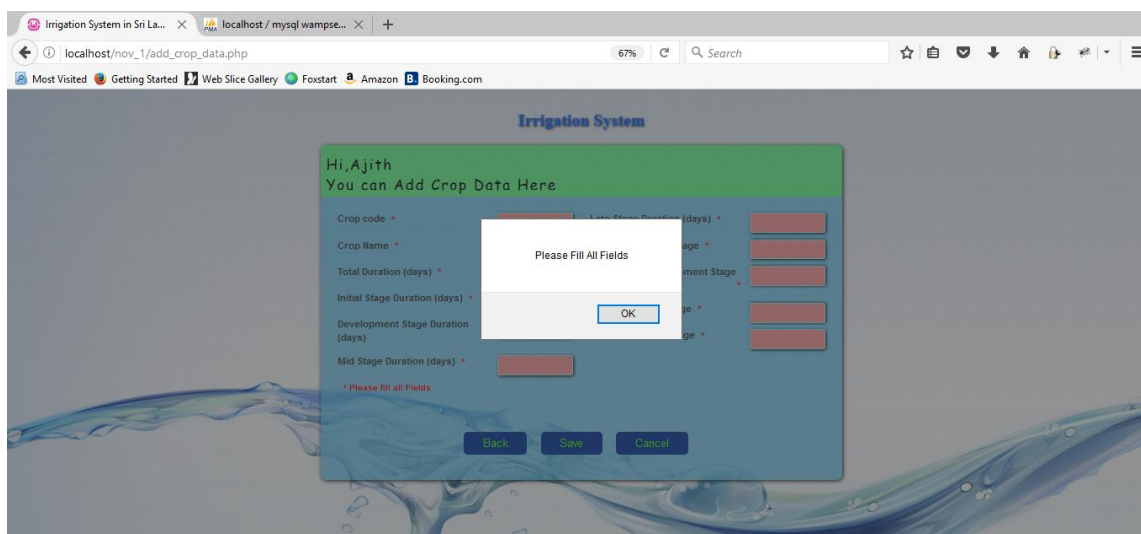


Figure A-14 Interface for adding the crop data, trying to save with required fields empty

When there is an attempt to save the data with blank required fields, or with a file name already exists, user will be notified the error, using the error message as shown in the *figure A-14* .

Figure A-15 below shows the message of invalid data input when user try to save the data without completing the all the fields with valid data through the reservoir-season interface. Data validation has been done in all the interfaces as much as possible.

Figure A-15 Response when adding invalid data through reservoir-season interface

Figure A-16, below shows the interface for adding the reservoir data. Before saving the data through this interface it is necessary to select the agro-ecological region and the evaporation station to which the reservoir belongs using the dropdown menu. It should be noted here that before adding a reservoir, evaporation data and agro-echo data should be added to the database.

Figure A-16 Interface for adding the reservoir data

Interface for adding the seasonal specific yield for a selected reservoir is shown in *figure A-17* below. Specific yield data can be added only after selecting the reservoir from the dropdown menu.

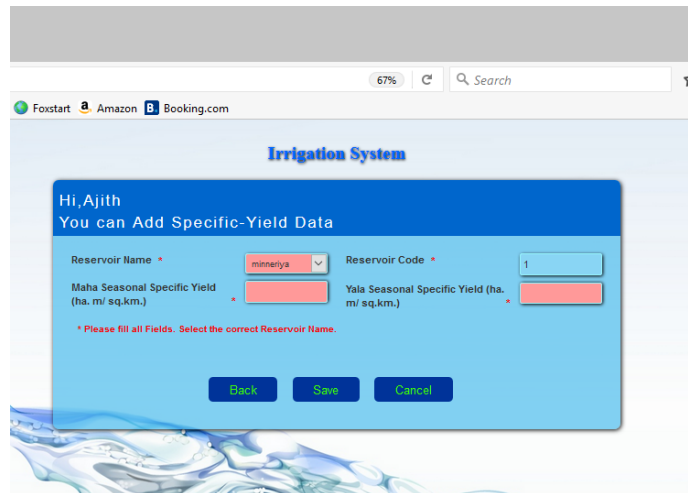







Figure A-17 interface for adding the seasonal specific yield data

A 6 DELETE DATA

User with successful logging, can select the option to delete the data. If the user feels that the data available in the database is not useful any more, such user can make use of this facility to save the computing resources. Data deletion facility is available only for the Admin, level 1 user, and the level 2 user only. This facility is not allowed for other two user levels. There are five interfaces through which the user can delete the data. They are listed below.

-  Delete rainfall data.
-  Delete evaporation data.
-  Delete crop data.
-  Delete reservoir-season data.
-  Delete crop-season data.

The privileged user can delete the data and go back to the main page if required. User has to select the base parameter on which the data to be deleted is tied on. Some of the interfaces are given in the figures below. Main page of delete data, after successful logging is shown in *figure A -18* below.

In data deletion, an alert message will pop up to confirm the action. At this moment, still the user can cancel the selection of deletion. Data will be deleted only after the user confirm the action. After deletion, the response message will appear to confirm the deletion with the key parameter of the deleted record.



Figure A-18 Main page to delete the data

The user can select any of the option from the main page of “delete data” at a time. Once user selects an option he will be navigated to the next relevant page. Figure A-19, below shows the interface for delete the rainfall data.

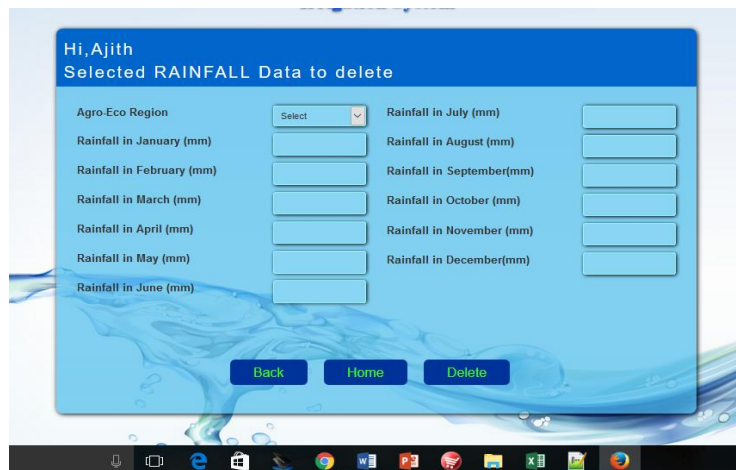


Figure A-19 Page to delete the rainfall data

Figure A-20 below shows the alert message before deletion that user should confirm to continue or can cancel the selection.

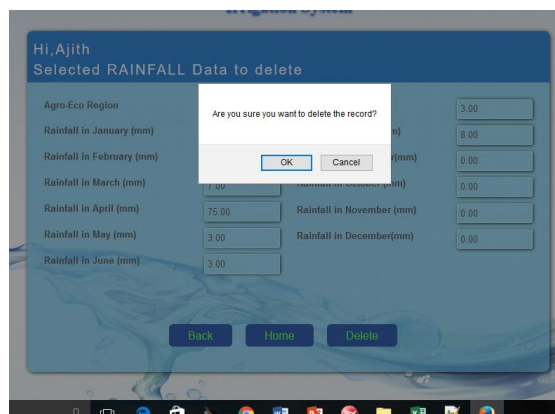


Figure A-20 Alert message before execute the deletion

Figure A-21 shows the response message after deletion. This will appear only after confirming the deletion in the interface shown in figure A-20.

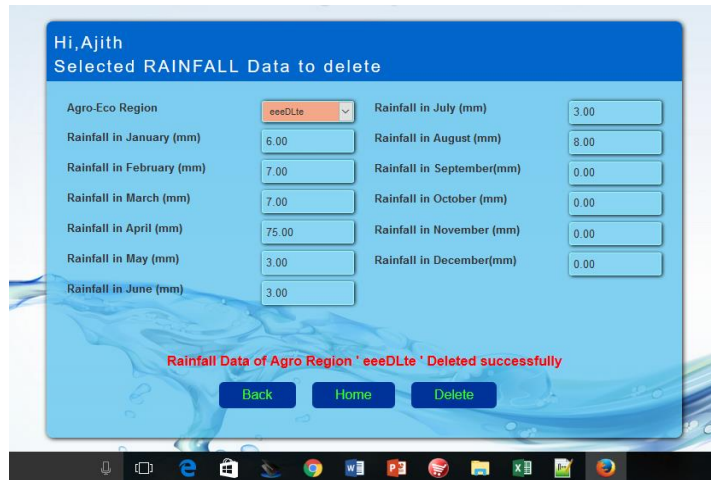


Figure A-21 system response message after deletion

Similar to the deletion of crop data, user can delete the data of other fields allowed by the system. Those are shown in the figure A-18, main page of delete data. Hence deletion interfaces of other fields are not shown in this manual.

A 7 EDIT DATA

Editing the existing data is a limited application in reservoir operation system. The main reason is that, there are files in “reservoir-season” module and “crop-season” module, which are created from the row data of other parameters such as rainfall, crop, evaporation, etc. Hence it is only possible to edit the limited data fields of the reservoir-season data and crop-season data such as Mahaweli inflow, efficiency, etc.

However, to be a complete software, editing facility is enabled in crop-season file and reservoir-season file, but it is necessary to save such files with a new file name. This ensures that the original files will remain as it is so the user can verify the previous results with original files. Figure A-22 below shows the interface to edit the reservoir-season data and crop-season data.

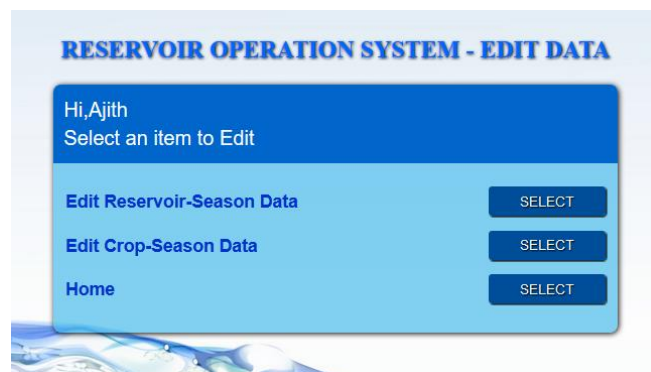


Figure A- 22 Main page to edit the data

Once the user selects the edit reservoir-season data in *figure A-22*, the interface shown in *figure A-23* will appear. With the editing facility, the user can retrieve the data of an existing file and change the values of Mahaweli inflow, Area to be cultivated, starting date of the season, efficiencies, initial storage, land preparation duration, etc. After changing one or more parameter, system will allow to save them only with a new file name. The relevant interfaces are discussed and shown below.

Hi, Admin
 Select Reservoir-Season file to Edit
 You can edit the data in light blue cells and save as a new file

File Name *	min_maha_1	New File Name	
Reservoir Name *	minneriya	Start Date of the Season (MM/DD/YY) *	11/19/2016
Catchment Area (Sq.km)	238.000	Start Date of the Season (MM/DD/YY) *	06/04/2017
Ago Eco Zone	DL1	Season Name	YALA
FSL (m. MSL)	93.700	EVP Station	kantalai
Capacity at FSL (MCM)	137.700	Seasonal Specific Yield (ha.m/sq.km.)	
MOL (m MCM)	85.900	Seasonal Rainfall (mm)	457.2
Minimum Storage (MCM)	12.140	Expected Rainfall Variation (%)	250.000
Irrigable Area (ha.)	10080.000	Expected Seasonal Rainfall (mm)	1968.800
storage at the beginning (MCM)	80.00	Percentage of Irrigable Area Forego (%)	30.000
		Area Cultivate in the Season (ha.)	7056.00

ENTER THE EXPECTED INFLOWS DURING THE SEASON

Month	APR	MAY	JUN	JUL	AUG	SEP	
Mahaweli Inflow(MCM)	20.00	10.00	20.00	10.00	5.00	10.00	
Inflow from ur's reservoir (MCM)	0.00	0.00	0.00	0.00	0.00	0.00	
Outflow to D/S reservoirs (MCM)	35	28	52	20	3	7	
Application Efficiency (%)	75					Conveyance Efficiency (%)	75

File Name Required

SAVE BACK Home Cancel

Figure A-23 Interface for edit and save the reservoir-season data

Above interface the existing file name, exiting date of commencement in pink color, Data which are not in dark blue cells can be edited and saved with a new file name. Editing the crop-season data is also similar to this one. The user can use these files for the operation execution.

A 8 OPERATION EXECUTION

After adding the data successfully, the user can execute the operation. This is simply the calculation of water demands, water inflows, losses and final storage of the reservoir on monthly basis for the selected season. The results will be presented in the table designed for the output. The graphical presentation is also available for better visualization. It is possible to get the print out of the final output.

Any registered user can carry out the operation execution process. *Figure A -24* shows the interface that appear when the user selects the *Execute Operation* button. Once this interface appeared, user is asked to select the reservoir-season file name and the crop-season filename. When user selects those two files which are really saved by the user under Add Data section using the, “create reservoir-season data” interface and the “create crop-season data” interfaces.

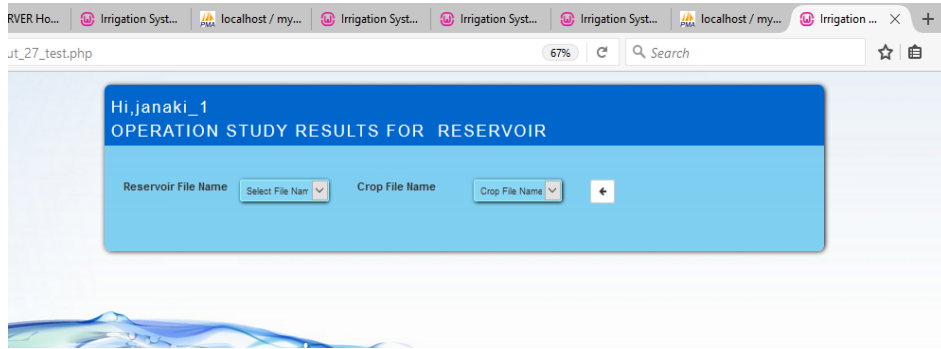


Figure A-24 Interface for execute the operation

Once user selects two file names where the data that the user wants to test are stored, the same interface expands to a table with required information. With the presentation of the results, majority of the functionalities of the reservoir operation system is over.

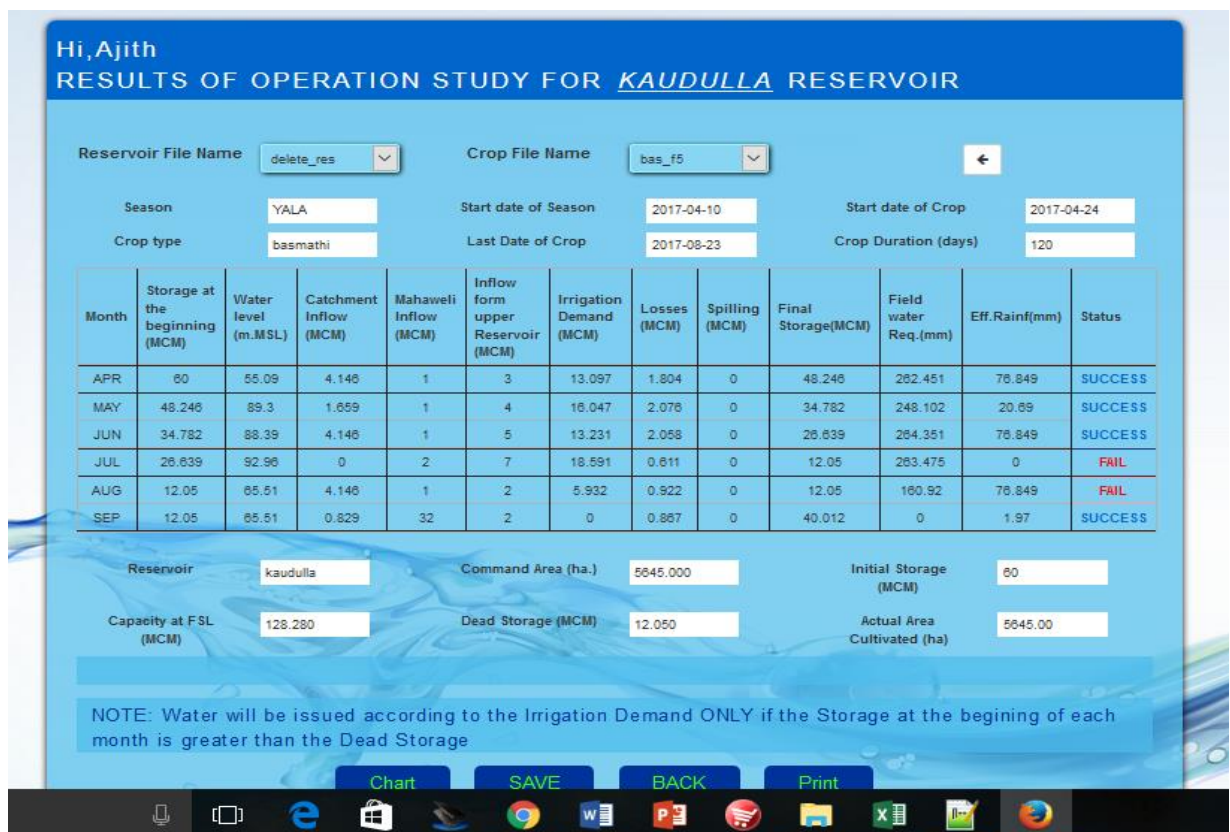


Figure A-25 Presentation of results after executing the programme

Figure A -25 above, shows the final output of the programme, It gives the Storage variation at the end of each moth together with the crop name, cultivated area, season start date, spilling,

catchment inflow, Mahaweli inflow, effective rainfall and the field water requirement, etc. Further it gives the success or the failure of the cultivation in the far right column in blue or red color respectively. It is expected that user will understand that user will adjust the inflows or other variables through the *edit* interface to make it success at the first month of failure and try the next trial. The user can try different combination of *crops* and *season start dates* by changing relevant details to get the most feasible combinations of start date of cultivation, monthly Mahaweli inflows, initial storage of the reservoir, land preparation durations and its water requirements, etc.

After obtaining the results, the user can navigate to view the results in graphical form as well. It is shown in the *figure A -26* below.

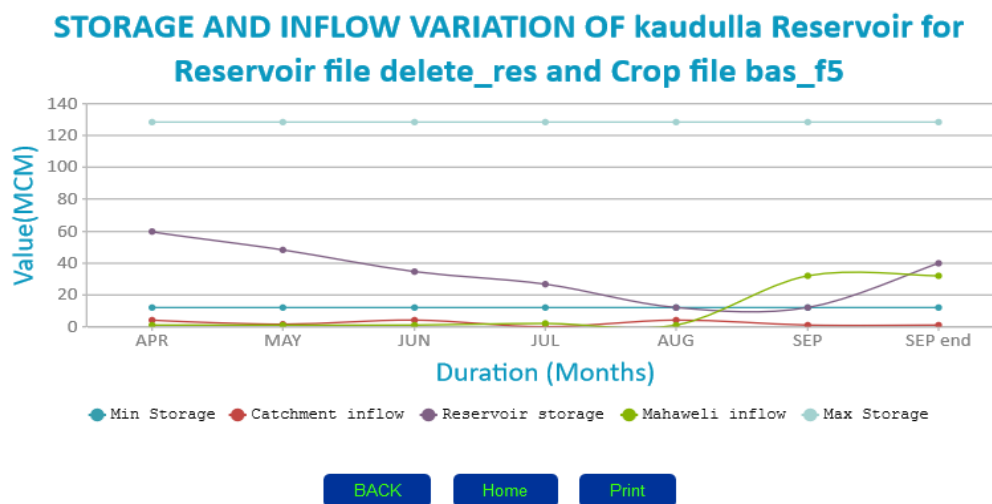


Figure A-26 Graphical presentation of key parameters of the final output

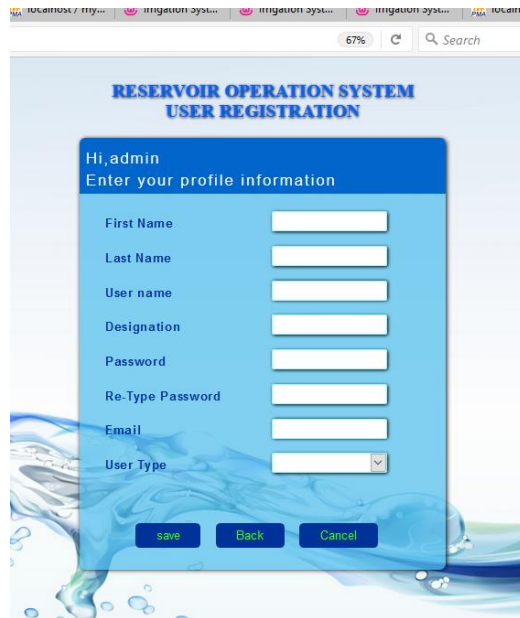
Using the back button in the graph the user can go back to the output interface. From the output interface, user can select a new execution or go back to home page by selecting the home button. Finally, the user can print the results of both output table and the graph by selecting the print button. Both of the printed outputs are shown in Appendix H.

All the instructions for a registered user of any level, to handle the reservoir operation system software is explained up to *figure A-26* in the user manual. Interfaces and functions with similar behavior are not repeatedly explained in the user manual.

User management functions, which are limited to the system Admin and probably to a special user, are discussed in the next part of the user manual.

A 9 USER REGISTRATION

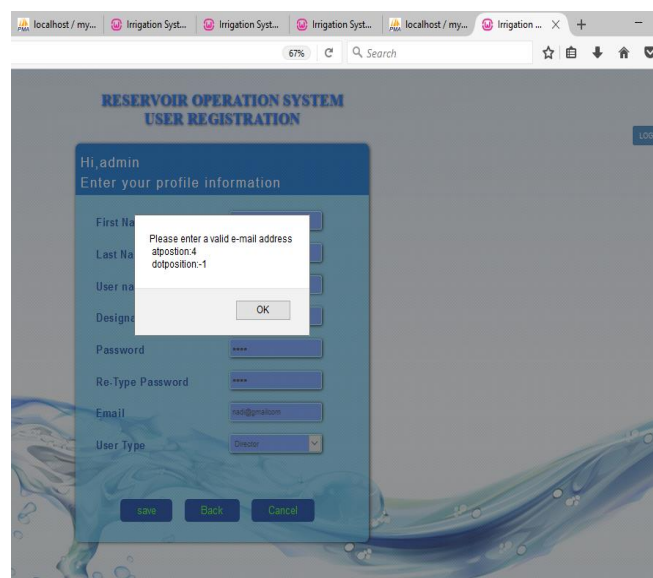
Only the System Admin can do the new user registration. The UI designed for the registration is shown in *figure A -27* below.



The screenshot shows a web browser window displaying the 'RESERVOIR OPERATION SYSTEM USER REGISTRATION' page. The page has a blue header with the text 'Hi, admin' and 'Enter your profile information'. Below the header is a form with the following fields: First Name, Last Name, User name, Designation, Password, Re-Type Password, Email, and User Type (a dropdown menu). At the bottom of the form are three buttons: 'save', 'Back', and 'Cancel'. The background of the page features a blue water splash graphic.

Figure A-27 interface for user registration

Once above UI appears, admin can complete the relevant information and save it under a specific user name. There are several checks embedded inside to ensure the password strength, validate the e-mail address, validate the confirm password, validate the user level and validate the first name and last name. Password validation and e-mail validation are shown in the *figure A-28* and *figure A -29* respectively with system responses.



The screenshot shows the same user registration form as in Figure A-27, but with a validation error. A white dialog box is displayed over the form with the text: 'Please enter a valid e-mail address' followed by 'ajposition.4' and 'ajposition-1'. An 'OK' button is visible in the dialog box. The form fields are partially obscured by the dialog box. The background of the page features a blue water splash graphic.

Figure A-28 e-mail validation in user registration

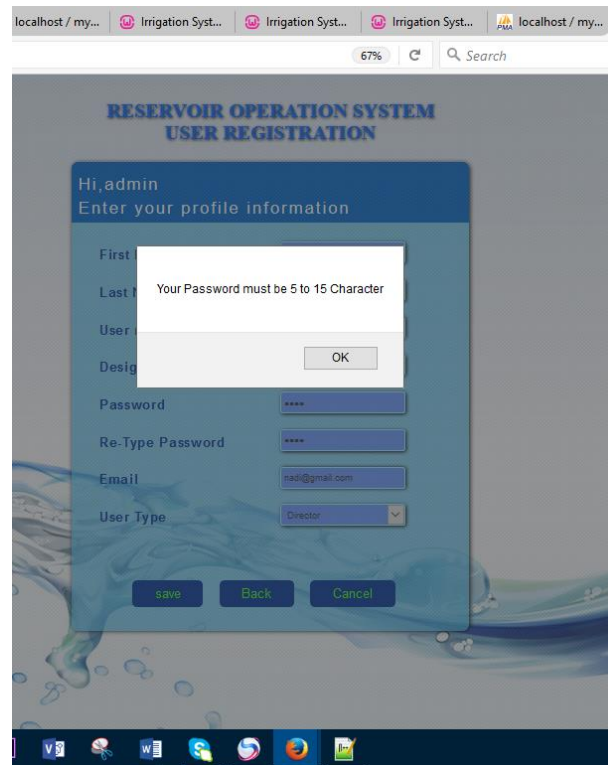


Figure A-29 Interface for password validation

After successful completion of the data, user registration can be done and registration is notified through the response message which is shown in the figure A -30 below.

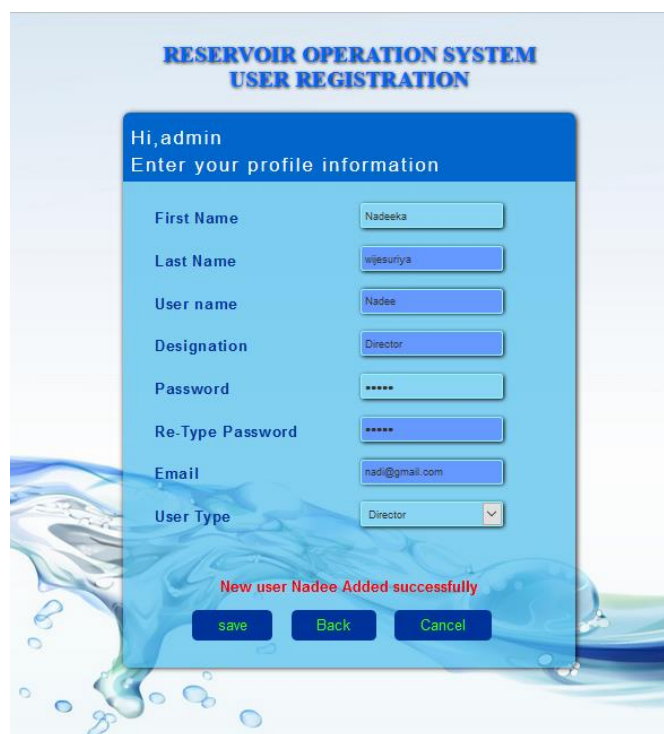


Figure A-30 System response after successful user registration

A 10 REMOVE REGISTERED USER

This facility is also restricted to the Admin. After logging as admin, it is possible to remove a user by selecting the identified user. This also has the checks to block the deletion of Admin himself and confirm to delete alert. *Figure A -31* shows the interface to select the user.

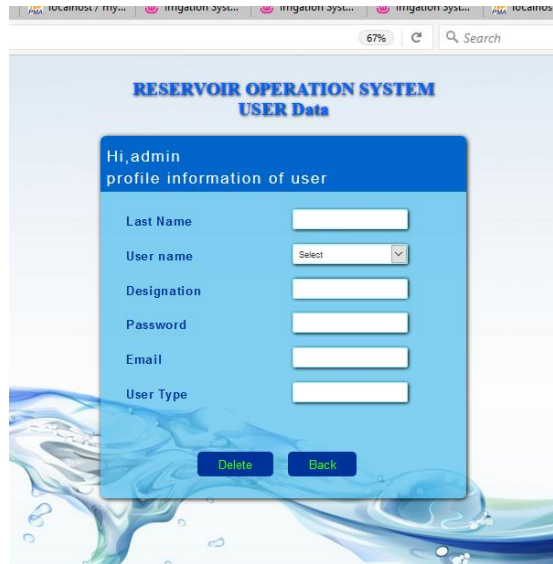


Figure A-31 Interface to select the user to be removed

Once the user is selected, and select to delete, system sends the alert message as shown in the *figure A -32* below. After confirming to delete by selecting OK button, user will be removed and the system gives the confirmation message as shown in *figure A-33*

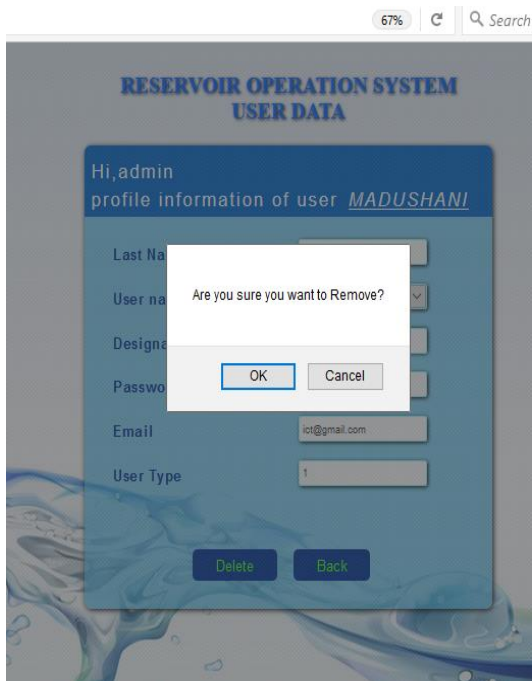


Figure A-32 Alert message to request the confirmation to delete

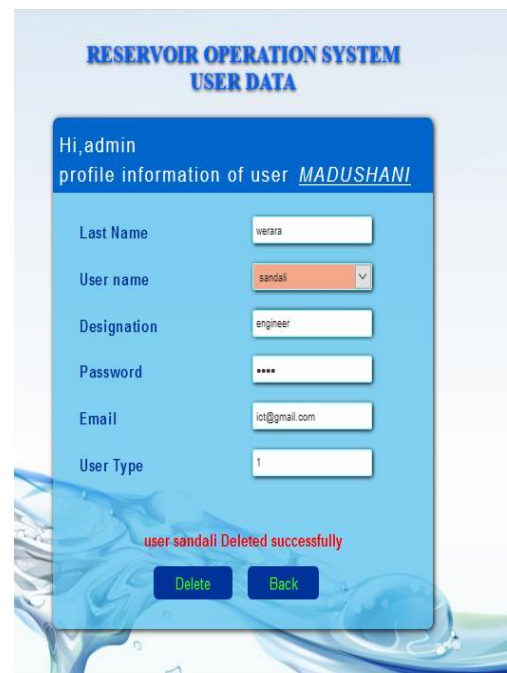


Figure A-33 Confirmation message after deletion

A 11 PASSWORD RESET

Any registered user can change his/her password by self, once logging. The same password validation is performed as done in the user registration here too. In addition to that, validation of current password is also done.

Figure A -34 below shows the interface for changing the password.

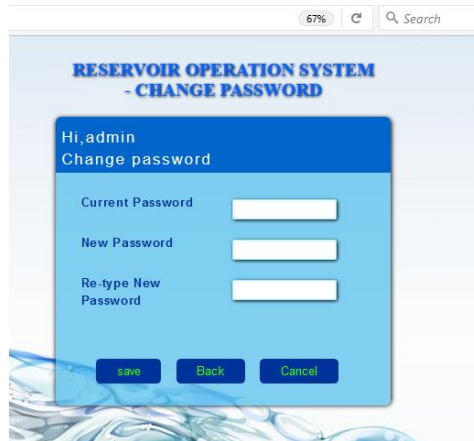


Figure A-34 interface to change the password

Figure A -35 shows the error message when entering the wrong password. After entering the correct password with validation satisfied, new password save message will appear as shown in figure A -36 below.

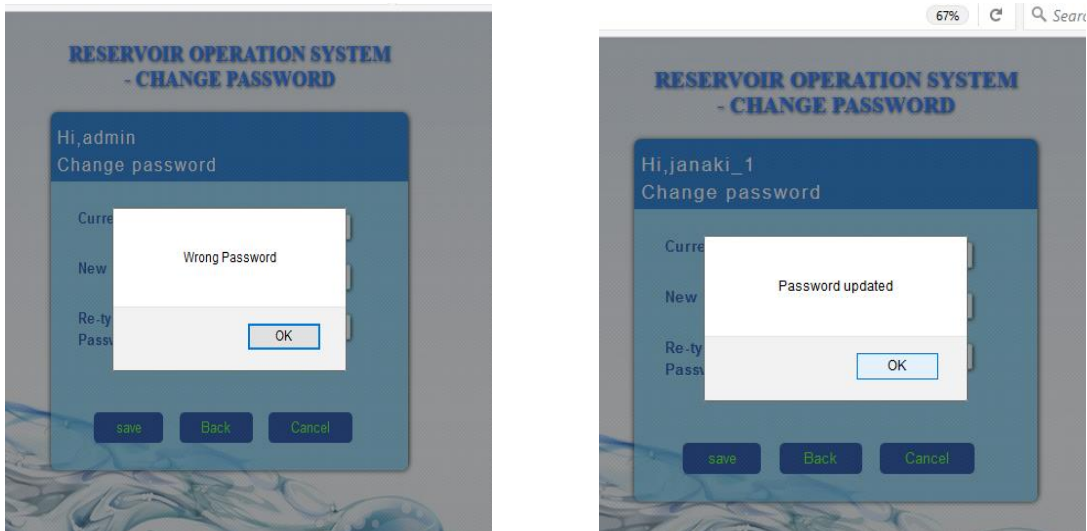


Figure A-35 Alert message in entering wrong pass

Figure A-36 Confirmation message in password reset

APPENDIX B DATABASE TABLE DETAILS

Table *B-1* to *B-8* are the basic tables where the data is stored from the outside. Table *B-9* and table *B-10* are the tables where the data is stored through AJAX functions using the reservoir-season interface and crop-season interface.

res_code	INT	4	PK, AUTO INCREMENT
res_name	VARCHAR	15	
cathm_area_sqkm	DEC	(9,3)	
Irr_area_ha	INT	4	
MOL_m_MSL	DEC	(9,3)	
FSL (m MSL)	DEC	(9,3)	
min_storage_MCM	DEC	(9,3)	
max_storage_MCM	DEC	(9,3)	
agro_eco_region	CHAR	4	
evapo_sta_name	VARCHAR	15	

Table B-1 Database table 'reservoir_data'

cap_id	INT	4	PK, AUTO INCREMENT
res_code	INT	4	FK, REFERENCES reservoir_data
Elevation_m_MSL	DEC	(8,3)	
cont_int_m	DEC	(6,3)	
Area_ha	DEC	(8,3)	
cum_cap_mcm	DEC	(6,3)	

Table B-2 Database table 'area_capacity'

crop_code	INT	4	PK, AUTO INCREMENT
crop_name	VARCHAR	15	
total_duration_days	INT	4	
dur_ini	INT	4	
dur_dev	INT	4	
dur_mid	INT	4	
dur_late	INT	4	
crop_fac_ini	DEC	(6,2)	
crop_fac_mid	DEC	(6,2)	
crop_fac_late	DEC	(6,2)	

Table B-3 Database table 'crop_varity'

Table No. 4 “evaporation”

evp_id	INT	4	PK, AUTO INCREMENT
sta_code	INT	4	
evapo_sta_name	VARCHAR	10	
month_name	CHAR	4	FK, REFERENCES 'season'
Evaporation_mm	DEC	(6,2)	

Table B-4 Database table 'evaporation'

sea_rec_id	INT	4	PK
season_code	INT	4	
season_name	CHAR	6	
month_number	INT	4	
month_name	CHAR	4	
month_days	INT	4	

Table B-5 Database table 'season'

iso_id	INT	4	PK, AUTO INCREMENT
season_code	INT	4	FK, , REFERENCES 'season'
res_code	INT	4	FK, REFERENCES 'reservoir_data'
iso_yield_mm	DEC	(8,3)	

Table B-6 Database table 'iso_yield'

rain_id	INT	4	PK, AUTO INCREMENT
agro_eco_region	CHAR	4	FK, REFERENCES 'reservoir_data'
month_name	CHAR	4	FK, REFERENCES 'season'
season_name	VARCHAR	255	
Rainfall_mm_	DEC	(6,2)	

Table B-7 Database table 'rainfall'

rec_id	INT	4	PK, AUTO INCREMENT
season_name	CHAR	4	FK, REFERENCES 'season'
month_name	CHAR	4	FK, REFERENCES 'season'
value_mmpday	DEC	(6,2)	

Table B-8 Database table 'ref_crop_evptr'

s_crop_code	INT	4	PK, AUTO INCREMENT
file_name	VARCHAR	15	
s_crop_dur	INT	4	
s_crop_name	VARCHAR	15	
s_dur_ini	INT	4	
s_dur_dev	INT	4	
s_dur_mid	INT	4	
s_dur_late	INT	4	
s_crop_fac_ini	DEC	(6,2)	
s_crop_fac_dev	DEC	(6,2)	

s_crop_fac_mid	DEC	(6,2)	
s_crop_fac_late	DEC	(6,2)	
s_lsd	INT	4	
s_ltd	INT	4	
s_lswr	INT	4	
s_ltwr	INT	4	

Table B-9 table crop_save, where data saved from crop-season UI

The table where the data saved from reservoir-season UI contains 39 fields, which is shown below.

save_id	INT	4	PK, AUTO INCREMENT
file_name	VARCHAR	15	
res_name	VARCHAR	20	
cathm_area	DEC	(9,3)	
aen	VARCHAR	8	
fsl	DEC	(9,3)	
cfsl	DEC	(9,3)	
iso_yield_mm	DEC	(6,2)	
sb	DEC	(9,3)	
season_d	DATE		
season_n	VARCHAR	15	
evp	DEC	(6,2)	
mol	DEC	(9,3)	
mins	DEC	(9,3)	
iah	DEC	(9,3)	
srv	DEC	(9,3)	
erv	DEC	(9,3)	
cal	DEC	(9,3)	
mon1	CHAR	6	
mon2	CHAR	6	
mon3	CHAR	6	
mon4	CHAR	6	
mon5	CHAR	6	

mon6	CHAR	6	
mi1	DEC	(6,2)	
mi2	DEC	(6,2)	
mi3	DEC	(6,2)	
mi4	DEC	(6,2)	
mi5	DEC	(6,2)	
mi6	DEC	(6,2)	
ifr1	DEC	(6,2)	
ifr2	DEC	(6,2)	
ifr3	DEC	(6,2)	
ifr4	DEC	(6,2)	
ifr5	DEC	(6,2)	
ifr6	DEC	(6,2)	
aeff	DEC	(3,2)	
ceff	DEC	(3,2)	
act_iah	DEC	(6,2)	

Table B-10 table save, where data saved from reservoir-season UI

APPENDIX C – TEST CASES

C 1 - TEST CASE FOR INVALID USER LOGGING

Test case ID -	2	Test Priority -	high		
Test case Title – Verify the rejection of unregistered user logging					
Pre- Condition – User must not be a registered person.					
Testing details					
Step	Description of step	Test data	Expected result	Actual result	Status (pass/fail)
1	Navigate to logging page	Username and password			
2	Input a username				
3	Input a password				
4	Click logging button		User should Not be able to log to the site	User is navigated with error message	PASS
Post-Condition - User is rejected and redirected to logging page.					

Table C-1 Test case for unregistered user logging.


C 2 SUMMARY LIST OF TEST CASES

Test cases relevant to user Management

Test ID	Test functionality	Procedure	Expected outcome	Status
5	Add user, without password	Admin selects the add user button, complete other fields except password field and click save	Alert message requesting the password	PASS

6	Add user, with password length less than 5 characters	Admin selects the add user button, complete all fields but password field only 4 chars and click save	Alert message password length not enough	PASS
7	Add user, with password and confirm password not same	Admin selects the add user button, complete all fields but password field and cpassword field are not same and click save	Alert message password and confirm password not matched	PASS
8	Add user, with blank user level	Admin selects the add user button, complete all fields but without selecting the user group and clicks on save	Alert message to select the user category	PASS
9	Add user, with existing username	Admin selects the add user button, complete all fields properly but filled the username existing username and clicks on save	Alert message user already exists	PASS
10	Add user, with invalid e-mail	Admin selects the add user button, complete all fields properly but filled the email field “dot” and/or “at” either missing or wrong locations and clicks on save	Alert message “email invalid”	PASS
11	Add user with user level as “Admin”	Admin selects the add user button, complete all fields properly but user level selected as Admin clicks on save	Alert message that”Admin level” cannot be added	PASS
12	Delete Admin	Admin selects delete user button, then selects Admin’s username and clicks delete button	Alert message that “Admin cannot be deleted”	PASS
13	Add a user	Admin selects add user, complete all the field correctly, avoiding above listed mistakes and clicks on save	Response message appear user” “ added successfully	PASS

Table C-2 test cases on user management

 **Test cases on adding crop data**

Test ID	Test functionality	Procedure	Expected outcome	Status
14	Add crop variety without crop name	User selects the add crop data , then complete all other fields except the crop name then clicks save	Alert message requesting the crop name	PASS
15	Add crop variety completing all fields, except any single field	User selects the add crop data, then complete all other fields including crop name but keep any single field blank and then clicks save. This step should repeat for 10 times as there are 10 fields	Alert message to fill the blank field and highlighting it.	PASS
16	Add crop variety completing all fields but total duration of the crop does not match with the sum of individual durations.	User selects the add crop data, then complete all the fields but enter a figure to total duration of the crop that does not match with the sum of individual durations.	Alert message crop duration does not match	PASS
17	Add crop variety completing all fields properly, avoiding above mistakes	User selects the add crop data, then complete all the fields properly. Crop name, crop code, crop duration match with the sum of individual durations.	Response message appear crop ” “ added successfully	PASS

Table C-3 Test cases on add crop data

 **Test cases on adding crop-season data**

Test ID	Test functionality	Procedure	Expected outcome	Status
18	Add crop_season file without filename	User selects the add crop_season file, then selects the relevant crop from the drop-down menu, complete all other field except the file name then clickssave	Alert message requesting the filename	PASS
19	Add crop_season file with existing filename	User selects the add crop_season data, then selects the relevant crop from the drop-down menu, complete all other field correctly except the file name which is already available then clicks save	Alert message filename already exists	PASS
20	Add crop_season file without selecting the crop.	User selects the add crop_season data, enter the file name to be saved, then clicks on save without selecting the crop.	Alert message crop must be selected.	PASS
21	Add crop_season file keeping any of the user input field blank	User selects the add crop_season data, then select the crop from drop-down, and then clicks on save tab while keeping a single field blank at a time. There are 4 such fields, so this step should repeat 4 times	Alert message “requesting to complete the empty field(s)” and highlight the empty field(s)	PASS

Table C-4 Test cases on add crop_season data

APPENDIX D – SAMPLE CODES

D 1 sample code for retrieving the data from database & save in table 'save'

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, maximum-scale=1">
<title>Irrigation System in Sri Lanka</title>
<!--<link rel="icon" href="favicon.png" type="image/png">-->
<link href="css/bootstrap.min.css" rel="stylesheet" type="text/css">
<link href="css/style.css" rel="stylesheet" type="text/css">
<link href="css/font-awesome.css" rel="stylesheet" type="text/css">
<script src="js/jquery-1.11.0.min.js"></script>
<script type="text/javascript">
$(document).ready(
function(){
$("#res_name").change(function() {
    //get the selected value
    var selectedValue = this.value;
    $("#res_name").css('background-color', '#89d5f4');
    var dataString2 = 'selectedValue='+ selectedValue;

    //make the ajax call
    $.ajax({
        url: 'http://localhost/nov_1/include/function.php',
        type: 'POST',
        data: dataString2,
        success: function(data) {
    var result = $.parseJSON(data);
    $('#cathm_area').val(result.cathm_area);
    $('#aen').val(result.aen);
    $('#fsl').val(result.fsl);
    $('#cfs1').val(result.cfs1);
    $('#evp').val(result.evp);
```



```

        $('#mol').val(result.mol);
        $('#mins').val(result.mins);
        $('#iah').val(result.iah);
    },
    error: function(data) {
        // Stuff
    }
});
});

$(".season_d").on("change paste keyup", function() {
$(".season_d").css('background-color', '#89d5f4');

var month = $(this).datepicker('getDate').getMonth();
this_month=month+1;
if(this_month==10 || this_month==11 || this_month==12 || this_month==1 ||
this_month==2 || this_month==3){

    $('#season_n').val("MAHA");
    var season_code="1";

    $('#mon1').val("OCT");
    $('#mon2').val("NOV");
    $('#mon3').val("DEC");
    $('#mon4').val("JAN");
    $('#mon5').val("FEB");
    $('#mon6').val("MAR");
}
else{

    $('#season_n').val("YALA");
    var season_code="2";

    $('#mon1').val("APR");
    $('#mon2').val("MAY");
    $('#mon3').val("JUN");
    $('#mon4').val("JUL");

```

```

    $('#mon5').val("AUG");
    $('#mon6').val("SEP");
    }
    //alert(season_code);
    var season_val = season_code;
var res_name_val= $("#res_name").val();
    //alert(res_name_val);
    var dataString3 = 'res_name_val=' + res_name_val + '&season_code='+
season_code;
    //make the ajax call
$.ajax({
    url: 'http://localhost/nov_1/include/season.php',
    type: 'POST',
    data: dataString3,
    success: function(data2) {
        var result2 = $.parseJSON(data2);
        $('#iso_yield_mm').val(result2.iso_yield_mm);
    },
    error: function(data2) {
// Stuff
}
    });
var aen_val = $("#aen").val();
var dataString4 = 'aen_val='+ aen_val + '&season_val=' + season_val;
$.ajax({
    url: 'http://localhost/nov_1/include/rainfall.php',
    type: 'POST',
    data: dataString4,
    success: function(data3) {
        $('#srv').val(data3);
    },
    error: function(data3) {
// Stuff
    }
    });
});
$("#erv, #srv").on("change paste keyup", function() {

```

```

var srv_val = $("#srv").val();
var erv_val = $("#erv").val();
var erv_cal_val = parseFloat(srv_val)*parseFloat(erv_val);
var erv_cal_val2 =parseFloat(erv_cal_val)/100;
var total = parseFloat(srv_val) + parseFloat(erv_cal_val2);
$("#cal").val(total);
});

$("#per_iah, #iah").on("change paste keyup", function() {
    var iah_val = $("#iah").val();
    var per_iah_val = $("#per_iah").val();
    var per_iah_cal_val = parseFloat(iah_val)*parseFloat(per_iah_val);
    var per_iah_cal_val2 =parseFloat(per_iah_cal_val)/100;
    var total_1 = parseFloat(iah_val) - parseFloat(per_iah_cal_val2);
    $("#act_iah").val(total_1);
});

$("#file_name").on("change paste keyup", function() {
    $("#file_name").css('background-color', '#89d5f4');
});

$("#saveid").click(function() {
    var name_status= $("#name_status").text();
    var file_name= $("#file_name").val();
    var res_name = $("#res_name").val();
    var cathm_area = $("#cathm_area").val();
    var aen = $("#aen").val();
    var fs1 = $("#fs1").val();
    var cfs1 = $("#cfs1").val();
    var iso_yield_mm = $("#iso_yield_mm").val();
    var sb = $("#sb").val();
    var season_d = $(".season_d").val();
    var season_n = $("#season_n").val();
    var evp = $("#evp").val();
    var mol = $("#mol").val();
    var mins = $("#mins").val();
    var iah = $("#iah").val();
    var per_iah=$("#per_iah").val();
    var act_iah = $("#act_iah").val();

```

```

var srv = $("#srv").val();
var erv = $("#erv").val();
var cal = $("#cal").val();
var mon1 = $("#mon1").val();
var mon2 = $("#mon2").val();
var mon3 = $("#mon3").val();
var mon4 = $("#mon4").val();
var mon5 = $("#mon5").val();
var mon6 = $("#mon6").val();
var mi1 = $("#mi1").val();
var mi2 = $("#mi2").val();
var mi3 = $("#mi3").val();
var mi4 = $("#mi4").val();
var mi5 = $("#mi5").val();
var mi6 = $("#mi6").val();
var ifr1 = $("#ifr1").val();
var ifr2 = $("#ifr2").val();
var ifr3 = $("#ifr3").val();
var ifr4 = $("#ifr4").val();
var ifr5 = $("#ifr5").val();
var ifr6 = $("#ifr6").val();
var aeff = $("#aeff").val();
var ceff = $("#ceff").val();

if(name_status=='File Name Already Exist'){
    $("#file_name").css('background-color','#f00');
    $("html, body").animate({ scrollTop: 0 }, "slow");
}
if(sb==''){$("#sb").css('background-color','#f00');
    $("html, body").animate({ scrollTop: 0 }, "slow");
}else{$("#sb").css('background-color','#89d5f4');
}
//alert(name_status);

if(file_name==''){$("#file_name").css('background-color','#f00');
    $("html, body").animate({ scrollTop: 0 }, "slow"); }else{

```

```

        if(res_name==''){$("#res_name").css('background-color','#f00');
        $("html, body").animate({ scrollTop: 0 }, "slow");
        }else{$("#res_name").css('background-color','#89d5f4');
        }
        if(season_d==''){$(".season_d").css('background-color','#f00');
        $("html, body").animate({ scrollTop: 0 }, "slow");
        }else{$("#res_name").css('background-color','#89d5f4');
        }
        if(mi1<0||mi2<0||mi3<0||mi4<0||mi5<0||mi6<0||ifr1<0||ifr2<0||ifr3<0||
        ifr4<0||ifr5<0||ifr6<0||mi6<0){
            alert("Negative inflows not accepted");
            exit();
        }
        var ret= $('.response').text();
var dataString5 = 'res_name=' + res_name + '&file_name=' + file_name +
'&cathm_area=' + cathm_area + '&aen=' + aen + '&fsl=' + fsl + '&cfs1=' +
cfs1 + '&iso_yield_mm=' + iso_yield_mm + '&sb=' + sb + '&season_d=' +
season_d + '&season_n=' + season_n + '&evp=' + evp + '&mol=' + mol +
'&mins=' + mins + '&iah=' + iah + '&act_iah='+ act_iah + '&srv=' + srv +
'&erv=' + erv + '&cal=' + cal + '&mon1=' + mon1 + '&mon2=' + mon2 +
'&mon3=' + mon3 + '&mon4=' + mon4 + '&mon5=' + mon5 + '&mon6=' + mon6 +
'&mi1=' + mi1 + '&mi2=' + mi2 + '&mi3=' + mi3 + '&mi4=' + mi4 + '&mi5=' +
mi5 + '&mi6=' + mi6 + '&ifr1=' + ifr1 + '&ifr2=' + ifr2 + '&ifr3=' + ifr3
+ '&ifr4=' + ifr4 + '&ifr5=' + ifr5 + '&ifr6=' + ifr6 + '&aeff=' + aeff +
'&ceff=' + ceff + '&per_iah=' + per_iah;

if(mi1==''||mi2==''||mi3==''||mi4==''||mi5==''||mi6==''||ifr1==''||ifr2=='
'||ifr3==''||ifr4==''||ifr5==''||ifr6==''||mi6==''||aeff==''||ceff==''||er
v==''||per_iah==''||sb==''){
    alert("Please Fill All Fields");
    exit();
}else
{
    $.ajax({
        type: "POST",
        url: "http://localhost/nov_1/include/save.php",
        data: dataString5,

```

```

        success: function(data5){
            $('<div id="response">'.response').text(data5);
            setTimeout(function() { $('<div id="response">'.response').text('')}, 100000 );
        }
    });
}

return false;
});
}); //doc.ready function close

```

```

function checkname()
{
    var name=document.getElementById( "file_name" ).value;
    if(name)
    {
        $.ajax({
            type: 'post',
            url: 'http://localhost/nov_1/include/checkdata.php',
            data: {
                file_name:name,
            },
            success: function (response) {
                $('#name_status').html(response);
                if(response=="File Name Success")
                {
                    return true;
                    //$('#file_name').css('background-color', '#89d5f4');
                }
                else
                {
                    return false;
                }
            }
        });
    }
    else
    {

```

```

$('#name_status').html("");
return false;
}
}
</script>
</head>

```

HTML part of the code is given below

```

<?php include 'include/config.php'; ?>
<?php
$sql = "SELECT * FROM reservoir_data";
$result = mysqli_query($conn,$sql);
?>
<header id="header_wrapper">
    <div class="mydiv"></div>
<h2 class="title">Reservoir & Seasonal data input</h2> </header>
<section id="water">
    <div class="container">
        <div class="transperent">
            <div class="head_home_res">First Select the Reservoir and then
Start date of the Season</div>
            <div class="input_form">
                <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
                    <form method="post" action="">
                        <div class="col-lg-6 col-md-6 col-sm-6 col-xs-6">
                            <div class="row">
                                <div class="lbl col-lg-8 col-md-8 col-sm-8 col-xs-8">
                                    <label class="labeln">File Name<span class="req">*</span></label>
                                </div>
                                <div class="tbox col-lg-4 col-md-4 col-sm-4 col-xs-4">
                                    <input type="text" class="tclass" name="file_name" id="file_name"
onkeyup="checkname();"><span id="name_status"></span>
                                </div>
                            <div class="clearfix"></div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

<label class="labeln">Reservoir Name<span class="req">*</span></label>
  <!--<span class="dis">(from res-data table)</span>-->
</div>
<div class="tbox col-lg-4 col-md-4 col-sm-4 col-xs-4">
  <select class="select" name="res_name1" id="res_name">
    <option value="">Select Reservoir Name</option>
    <?php
    while($row = mysqli_fetch_array($result))
    {
    echo ("<option
    value='".$row['res_code']."'>".$row['res_name']."</option>");
    }
    ?>
  </select>
</div>
<div class="clearfix"></div>
</div>
<div class="row">
  <div class="lbl col-lg-8 col-md-8 col-sm-8 col-xs-8">
  <label class="labeln">Catchment Area</label>
  </div>
  <div class="tbox col-lg-4 col-md-4 col-sm-4 col-xs-4">
  <input type="text" class="tclass" id="cathm_area">
  <span class="lab_n">KM 2</span></div>
  <div class="clearfix"></div>
</div>
<div class="row">
  <div class="lbl col-lg-8 col-md-8 col-sm-8 col-xs-8">
  <label class="labeln">Ago Eco Zone</label>
  </div>
  <div class="tbox col-lg-4 col-md-4 col-sm-4 col-xs-4">
  <input type="text" class="tclass" id="aen">
  </div>
<div class="clearfix"></div>
</div>
<div class="row">
  <div class="lbl col-lg-8 col-md-8 col-sm-8 col-xs-8">

```



```

<label class="labeln">FSL</label>
</div>
<div class="tbox col-lg-4 col-md-4 col-sm-4 col-xs-4">
<input type="text" value="readonly" class="tclass" id="fsl">
<span class="lab_n">M.MSL</span></div>
<div class="clearfix"></div>
</div>
<div class="row">
<div class="lbl col-lg-8 col-md-8 col-sm-8 col-xs-8">
<label class="labeln">Capacity at FSL</label>
</div>
<div class="tbox col-lg-4 col-md-4 col-sm-4 col-xs-4">
<input type="text" class="tclass" id="cfs1">
<span class="lab_n">MCM</span></div>
<div class="clearfix"></div>
</div>
<div class="row">
<div class="lbl col-lg-8 col-md-8 col-sm-8 col-xs-8">
<label class="labeln">MOL</label>
</div>
<div class="tbox col-lg-4 col-md-4 col-sm-4 col-xs-4">
<input type="text" class="tclass" id="mol">
<span class="lab_n">M.MSL</span></div>
<div class="clearfix"></div>
</div>
<div class="row">
<div class="lbl col-lg-8 col-md-8 col-sm-8 col-xs-8">
<label class="labeln">Minimum Storage</label>
</div>
<div class="tbox col-lg-4 col-md-4 col-sm-4 col-xs-4">
<input type="text" class="tclass" id="mins">
<span class="lab_n">MCM</span></div>
<div class="clearfix"></div>
</div>
<div class="row">
<div class="lbl col-lg-8 col-md-8 col-sm-8 col-xs-8">
<label class="labeln">Irrigable Area</label>

```

```

</div>
<div class="tbody col-lg-4 col-md-4 col-sm-4 col-xs-4">
<input type="text" class="tclass" id="iah">
<span class="lab_n">ha</span></div>
<div class="clearfix"></div>
</div>
  <div class="row">
<div class="lbl col-lg-8 col-md-8 col-sm-8 col-xs-8">
<label class="labeln">Storage at the begining</label>
</div>
<div class="tbody col-lg-4 col-md-4 col-sm-4 col-xs-4">
<input type="text" class="tclass" id="sb">
<span class="lab_n">MCM</span></div>
<div class="clearfix"></div>
</div>
  </div>
<div class="col-lg-6 col-md-6 col-sm-6 col-xs-6">

  <div class="row">
<div class="lbl col-lg-8 col-md-8 col-sm-8 col-xs-8">
<label class="labeln">Start Date of the Season<span
class="req">*</span></label>
</div>
<div class="tbody col-lg-4 col-md-4 col-sm-4 col-xs-4">
<input type="text" class="tclass datepicker season_d" id=""
placeholder="DD/MM/YYYY">
</div>
<div class="clearfix"></div>
</div>
<div class="row">
<div class="lbl col-lg-8 col-md-8 col-sm-8 col-xs-8">
<label class="labeln">Season Name </label>
</div>
<div class="tbody col-lg-4 col-md-4 col-sm-4 col-xs-4">
<input type="text" class="tclass" id="season_n">
</div>
<div class="clearfix"></div>

```

```

</div>
<div class="row">
  <div class="lbl col-lg-8 col-md-8 col-sm-8 col-xs-8">
    <label class="labeln">EVP Station</label>
  </div>
  <div class="tbox col-lg-4 col-md-4 col-sm-4 col-xs-4">
    <input type="text" class="tclass" id="evp">
  </div>
<div class="clearfix"></div>
</div>
<div class="row">
  <div class="lbl col-lg-8 col-md-8 col-sm-8 col-xs-8">
    <label class="labeln">Seasonal Specific Yield</label>
  </div>
  <div class="tbox col-lg-4 col-md-4 col-sm-4 col-xs-4">
    <input type="text" class="tclass" id="iso_yield_mm">
    <span class="lab_n">Ham/sq.km</span></div>
  <div class="clearfix"></div>
</div>
<div class="row">
  <div class="lbl col-lg-8 col-md-8 col-sm-8 col-xs-8">
    <label class="labeln">Seasonal Rainfall</label>
  </div>
  <div class="tbox col-lg-4 col-md-4 col-sm-4 col-xs-4">
    <input type="text" class="tclass" id="srv">
    <span class="lab_n">mm</span></div>
  <div class="clearfix"></div>
</div>
<div class="row">
  <div class="lbl col-lg-8 col-md-8 col-sm-8 col-xs-8">
    <label class="labeln">Expected Rainfall Variation</label>
  </div>
  <div class="tbox col-lg-4 col-md-4 col-sm-4 col-xs-4">
    <input type="text" class="tclass" id="erv">
    <span class="lab_n">%</span></div>
  <div class="clearfix"></div>
</div>

```

```

    <div class="row">
      <div class="lbl col-lg-8 col-md-8 col-sm-8 col-xs-8">
<label class="labeln">Expected Seasonal Rainfall</label>
      </div>
      <div class="tbox col-lg-4 col-md-4 col-sm-4 col-xs-4">
        <input type="text" class="tclass" id="cal" disabled>
        <span class="lab_n">%</span></div>
      <div class="clearfix"></div>
    </div>
    <div class="row">
      <div class="lbl col-lg-8 col-md-8 col-sm-8 col-xs-8">
<label class="labeln">Percentage of Irrigable Area Forego</label>
      </div>
      <div class="tbox col-lg-4 col-md-4 col-sm-4 col-xs-4">
        <input type="text" class="tclass" id="per_iah">
        <span class="lab_n">%</span></div>
      <div class="clearfix"></div>
    </div>
    <div class="row">
      <div class="lbl col-lg-8 col-md-8 col-sm-8 col-xs-8">
<label class="labeln">Area Cultivate in the Season</label>
      </div>
      <div class="tbox col-lg-4 col-md-4 col-sm-4 col-xs-4">
        <input type="text" class="tclass" id="act_iah">
        <span class="lab_n">%</span></div>
      <div class="clearfix"></div>
    </div>
  </div>
  <div class="clearfix"></div>
</form>
</div>
  <div class="clearfix"></div><h3 class="center sect">ENTER THE
EXPECTED INFLOWS DURING THE SEASON</h3>
<div class="corrospoding col-lg-12 col-md-12 col-sm-12 col-xs-12" >
  <ul class="months">
    <li class="tcol" style="align:center;">Month</li>
    <li>

```

```

<input type="text" name="month" id="mon1" class="tclassm">
  </li>
  <li>
    <input type="text" name="month" id="mon2" class="tclassm">
      </li>
      <li>
        <input type="text" name="month" id="mon3" class="tclassm">
          </li>
          <li>
            <input type="text" name="month" id="mon4" class="tclassm">
              </li>
              <li>
                <input type="text" name="month" id="mon5" class="tclassm">
                  </li>
                  <li>
                    <input type="text" name="month" id="mon6" class="tclassm">
                      </li>
                      <div class="clearfix"></div>
                    </ul>
                  </div>
                <div class="clearfix"></div>
              <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
                <div class="eff col-lg-3 col-md-3 col-sm-3 col-xs-3 paddingleft0">
                  <p>Application Efficiency</p>
                </div>
                <div class="eff col-lg-3 col-md-3 col-sm-3 col-xs-3">
                  <input type="text" name="eff1" class="tclass" id="aeff">
                  <span class="lab_n">%</span> </div>
                <div class="eff col-lg-3 col-md-3 col-sm-3 col-xs-3">
                  <p>Conveyance Efficiency</p>
                </div>
                <div class="eff col-lg-3 col-md-3 col-sm-3 col-xs-3">
                  <input type="text" name="eff2" class="tclass" id="ceff">
                  <span class="lab_n">%</span> </div>
                <div class="clearfix"></div>
              </div>
            <div class="clearfix"></div>
          </div>
        <div class="clearfix"></div>

```

```

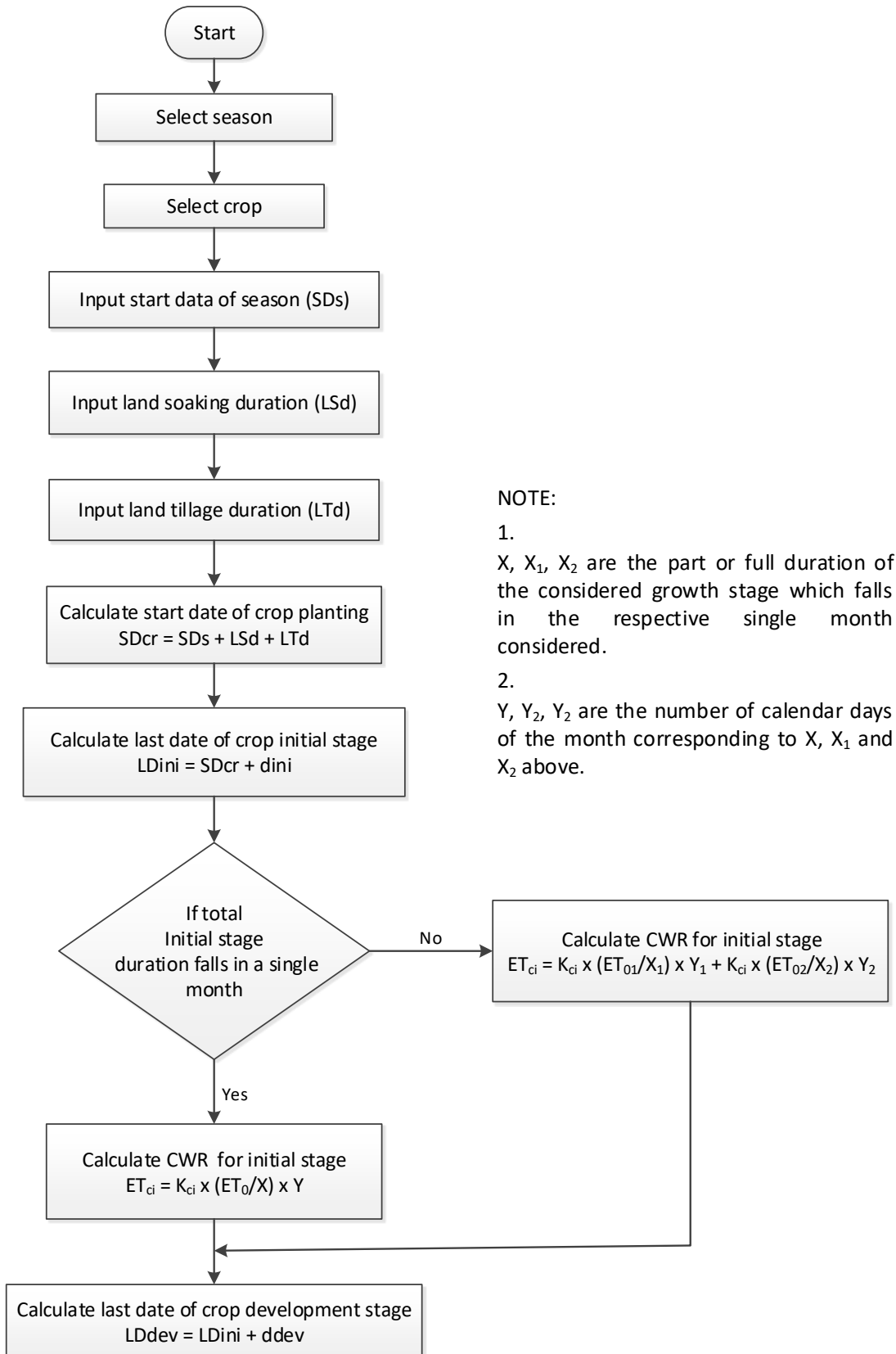
    <div class="btn_set">
    <div class="response"></div>
        <input type="submit" class="btns" id="saveid" value="SAVE">
    <a href="add_all.php"><input class="btns" id="back" value="BACK"
type="button"></a>
        <input type="submit" class="btns" id="clearid" value="CANCEL">
            <div class="clearfix"></div>
        </div>    </div> </div> </div> </div>
    <!-------container----->
</section>
<div class="footer"> </div>
<link rel="stylesheet"
href="http://code.jquery.com/ui/1.12.0/themes/base/jquery-ui.css">
<script src="http://code.jquery.com/ui/1.12.0/jquery-ui.js"></script>
<script>
    $( function() {
        $( ".datepicker" ).datepicker();
    } );
</script>
</body>
</html>

```

Data insert code is not shown here. Once the save_id is clicked, data will be posted to ../include/save.php and it will do the rest.

APPENDIX E - - CALCULATION FLOW CHART

E1 flow chart for calculating the crop water requirement



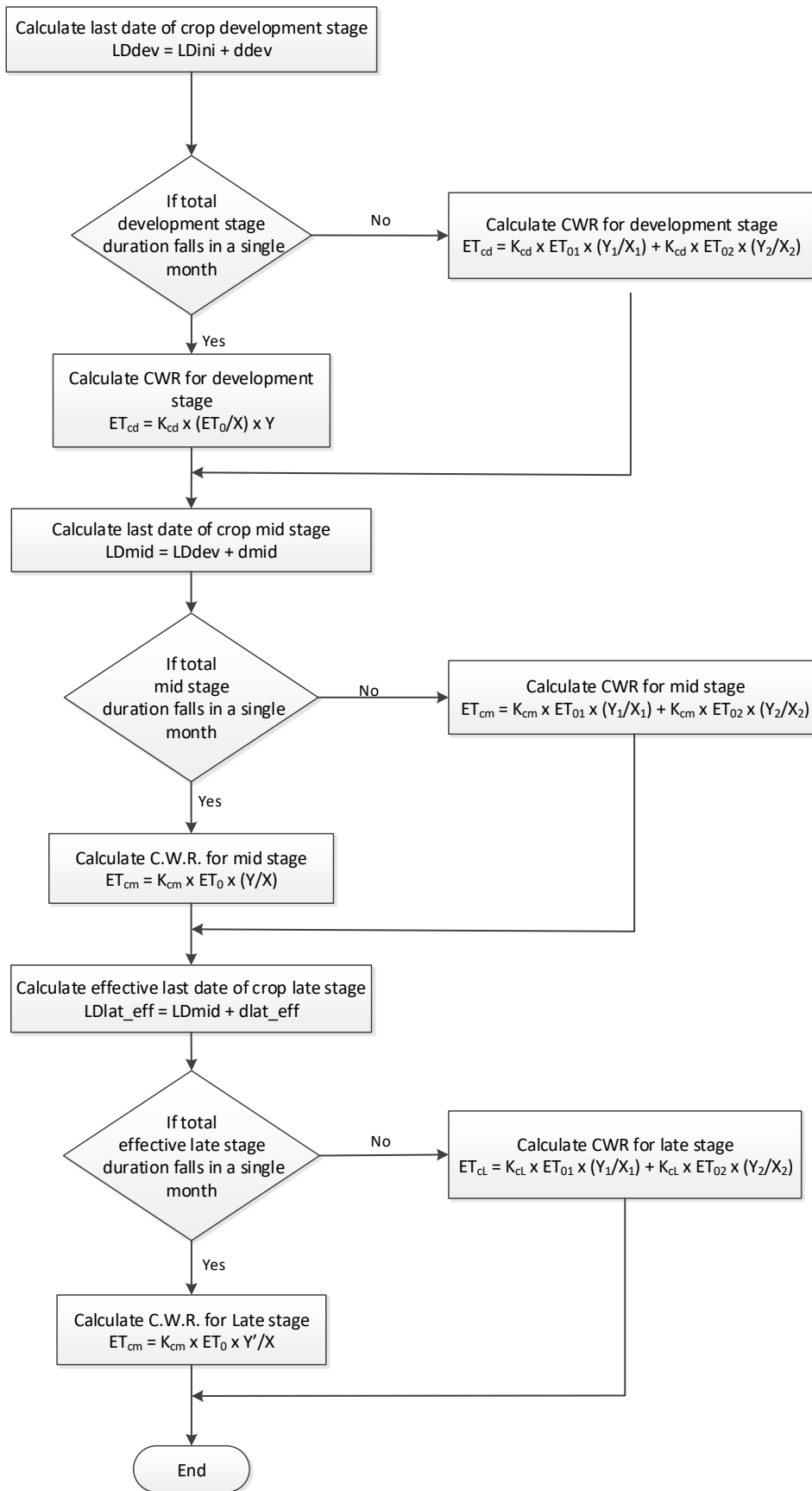


Figure E-1 Flow chart for calculating the crop water requirement for entire growth stages

APPENDIX F - RESERVOIR SYSTEM LAYOUT

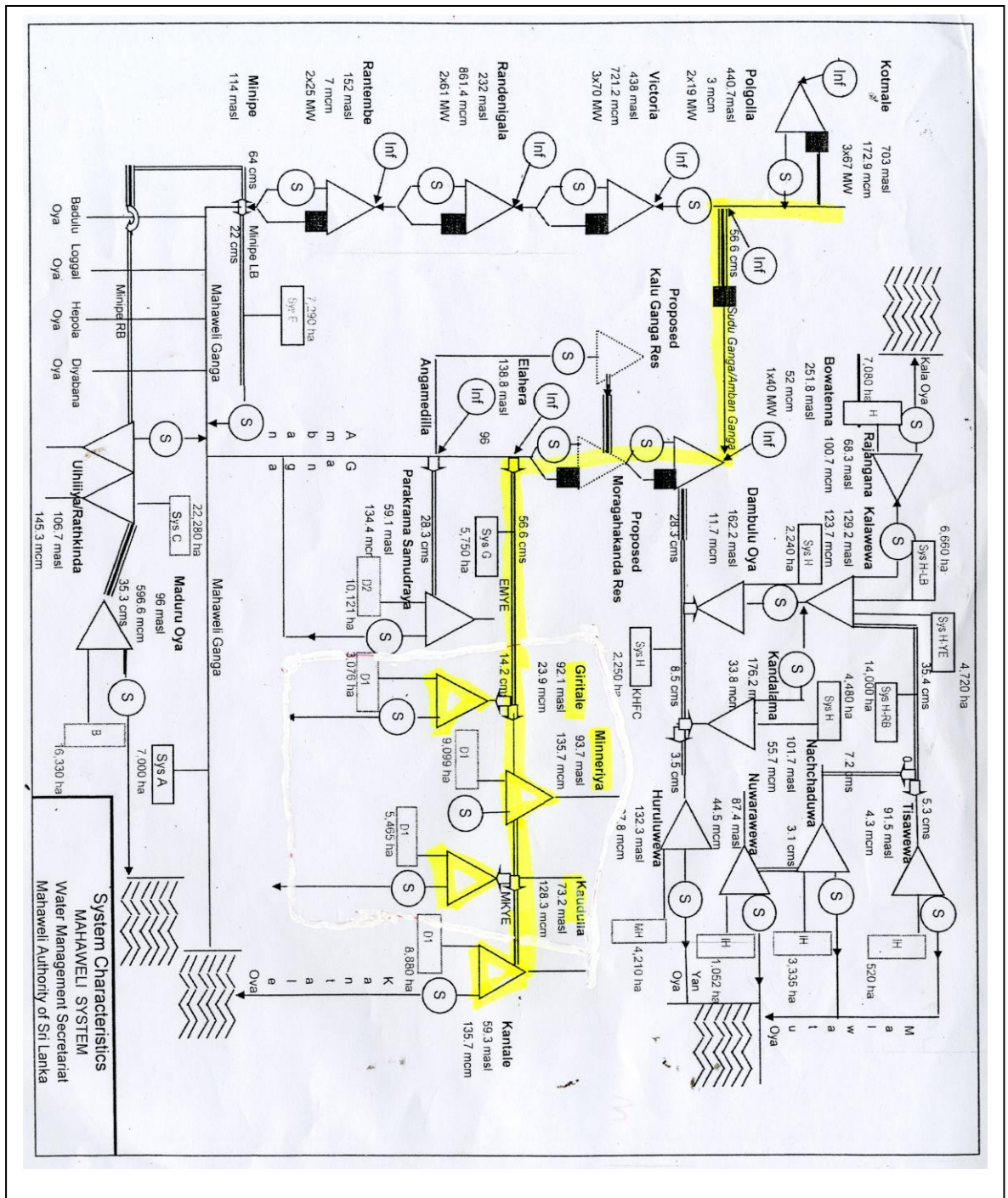


Figure F -1 Mahaweli water distribution system with selected reservoirs.

APPENDIX G – INITIAL DOCUMENTS

G 1 conceptual user interfaces

There are three most important and guided documents which act as the foundation for the success of the project. Those are little bit modified while carrying out the project. However, those are still useful for the comparison and justification purpose.

Reservoir name <input type="text"/> <input type="button" value="v"/> (from res-data table drop down)	Start date of the season <input type="text"/> <input type="button" value="v"/> <input type="text"/> <input type="button" value="v"/> DD / MM drop down
Catchment area <input type="text"/> km2 (automatically after selecting Res_name)	Season <input type="text"/> (automatically after selecting the Reservoir)
Ago Eco Zone <input type="text"/> (automatically after selecting the Reservoir)	MOL <input type="text"/> M.MSL (automatically once selecting the Reservoir)
EVP Station <input type="text"/> (From database)	Minimum Storage <input type="text"/> MCM (automatically once selecting the Reservoir)
FSL <input type="text"/> M.MSL (automatically once selecting the Reservoir)	Irrigable Area <input type="text"/> ha (automatically once selecting the Reservoir)
Capacity at FSL <input type="text"/> MCM (automatically once selecting the Reservoir)	Seasonal Rainfall <input type="text"/> mm (once the season and res. is selected this has to be calculated & displayed)
Seasonal Specific yield <input type="text"/> Ha m /sq.km (automatically after selecting the Reservoir)	Expected rainfall variation <input type="text"/> % (+25 -25)
Storage at the beginning <input type="text"/> MCM (Keyboard input)	Seasonal Rainfall <input type="text"/> % (+10 100)

Corresponding months to be displayed according to the season

Month	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Mahaweli inflow (MCM)	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Inflow from u/s reservoir	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Application efficiency <input type="text"/> %	Conveyance efficiency <input type="text"/> %
---	--

Figure G -1 Initial paper prototype to add reservoir-season data

Paddy Duration days
select from database

Paddy Variety
select from database

Initial stage duration days
(automatically once the variety is selected)

Development stage duration
(automatically once the variety is selected)

mid stage duration days
(automatically once the variety is selected)

Late stage duration days
(automatically once the variety is selected)

Kc initial
(Auto once Variety is select)

Kc dev
(Auto once Variety is select)

Kc mid
(Auto once Variety is select)

Kc late
(Auto once Variety is select)

land soaking duration days

Land tillage duration days

Land soaking water reqt mm

Land tillage water reqt mm

Figure G-2 Initial paper prototype to add crop-season data

APPENDIX H – USER CERTIFICATES



IESL College of Engineering

Engineering Your Future

A FULLY OWNED SUBSIDIARY OF THE INSTITUTION OF ENGINEERS OF SRI LANKA (IESL)
(IESL IS INCORPORATED BY ACT NO 17 OF 1968)

21st February 2017.

To whom it may concern

This note is submitted to endorse the validity and applicability of the final year project submitted by Mr. Ajith Gunasekara on "Reservoir operation system" for the partial fulfilment of the requirements of the MIT program in School of Computing of the University of Colombo.

The computer model developed by him is a very effective operational tool for carrying out reservoir operation for designing of Irrigation systems under different design scenarios and conditions. Facility provided in the software for different crop varieties with user opted cultivation starting date is a very user-friendly tool for the authorities in arriving at an optimal decision.

Further, the model is developed in a very user-friendly manner for anyone who wishes to carry out such studies. Although the procedure for carrying out such designs are given in the Irrigation Department guidelines, a user-friendly computer model has not been developed so far to get hands-on experience on such design works. The model developed by Mr. Gunasekara is very good teaching tool for capacity building programmes in the field of irrigation.

Mr. Ajith Gunasekera served in Irrigation Training Institute of the Irrigation Department as a lecturer while I was Director- In charge of the Institute. He was capable of developing many teaching tools using computer software and used at that time for teaching purposes. He served as Director of the Institute several years later and still continue to do such works as very senior level engineer for the benefit of the Irrigation Department.

As such I recommend computer model developed by him for the use in the Irrigation Department in their routine design works as well as in preparation of the scheme wise seasonal water requirements in a more transparent manner when requesting the water quota from the decision makers of the Mahaweli Water Management Secretariat.

Dr. (Eng.) G.G.A. Godaliyadda

B.Sc. Eng (Hons), M.Phil., Ph.D (Cornell, USA), C.Eng. FIE (SL), MICE (London)

Head, Civil Engineering Department,

IESL College of Engineering.

Former Director General of Irrigation

APPENDIX I – FINAL RESULTS

The results of the reservoir operation system is presented in tabular form and in graphical form both of which can be printed. The printing facility will be very much useful in the reporting and the higher level discussions where the software itself may not be needed to operate. The scanned images of the results are shown in *figure H-1* and *figure H-2* below.

Hi, Ajith

RESULTS OF OPERATION STUDY FOR MINNERIYA RESERVOIR

Reservoir File Name: Crop File Name:

Season: YALA Start date of Season: 2017-04-25 Start date of Crop: 2017-05-07

Crop type: basmathi Last Date of Crop: 2017-09-05 Crop Duration (days): 120

Month	Storage at the beginning (MCM)	Water level (m.MSL)	Catchment Inflow (MCM)	Mahaweli Inflow (MCM)	Inflow from upper Reservoir (MCM)	Irrigation Demand (MCM)	Losses (MCM)	Spilling (MCM)	Final Storage(MCM)	Field water Req.(mm)	Eff.Rainf(mm)	Status
APR	136.9	93.87	1.814	6	0	2.609	3.348	0	118.756	66.479	47.068	SUCCESS
MAY	118.756	72.83	0.725	11	0	52.704	3.776	0	55.002	400.923	8.777	SUCCESS
JUN	55.002	45	1.814	11	0	35.737	0.373	0	31.706	312.964	47.068	SUCCESS
JUL	31.706	67.95	0	4	0	42.063	1.812	0	12.14	312.966	0	FAIL
AUG	12.14	87.78	1.814	2.5	0	30.868	0.418	0	12.14	276.739	47.068	FAIL
SEP	12.14	87.78	0.363	0	0	3.658	0.395	0	12.14	27.214	0	FAIL

Reservoir: minneriya Command Area (ha.): 10080.000 Initial Storage (MCM): 136.9

Capacity at FSL (MCM): 137.700 Dead Storage (MCM): 12.140 Actual Area Cultivated (ha): 10080.00

NOTE: Water will be issued according to the Irrigation Demand ONLY if the Storage at the beginning of each month is greater than the Dead Storage

Figure I-3 Results of the Operation study in tabular form

In the results shown in *figure I-1*, all the important information such as, start date of the cultivation, area cultivated, crop variety, storages at the beginning of each month, irrigation requirements for each month, etc. are given,

Graph of the important parameters of the results shown in *figure I-1* is shown in the *figure I-2* below

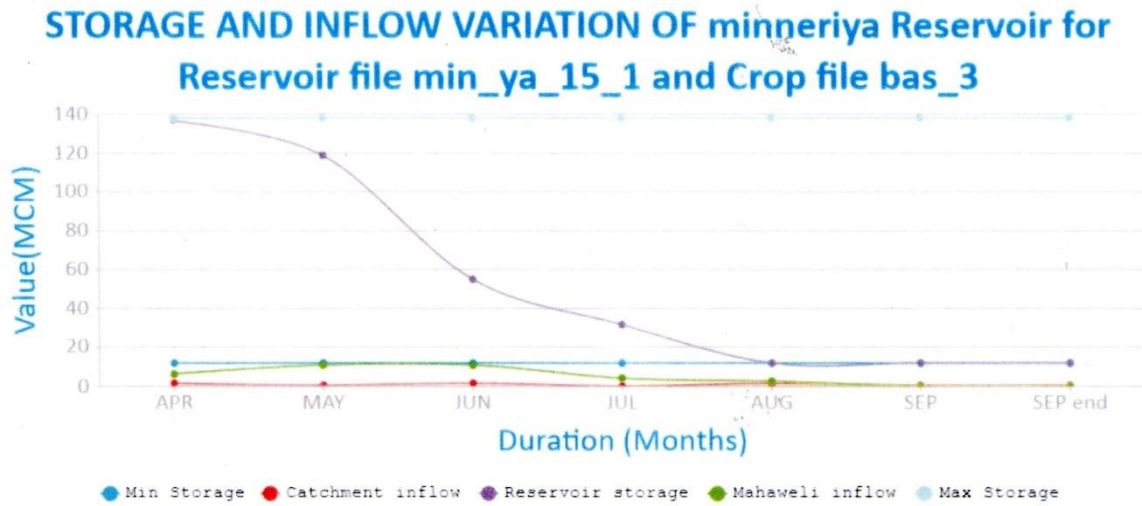


Figure I-4 Results in graphical form

Only the important parameters are shown in the graphical form of the results. This will enable the user to identify the parameters at a glance that should be adjusted to make the cultivation successful in case of cultivation failure or on the other hand to adjust the Mahaweli inflows or initial storage to prevent the spilling.