American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science & Technology (FST)

PROJECT TITLE

Automated Irrigation System that automates and controls the irrigation of crops based on soil moisture levels and weather conditions.

Submitted By:

Semester: Summer_21_22		**Section:**	**Group Number:**	
SN	Student Name	Student ID	Contribution (CO1+CO2)	Individual Marks
01	KHONDOKER MD. SABIT HASAN	21-45306-2	25%	
02	NOKIBUL ARFIN SIAM	21-44793-1	25%	
03	ABU NASER MD. ARMAN	21-45239-2	20%	
04	FAHIM RAHMAN	21-44399-1	15%	
05	MD. NAJIB HOSSAIN	21-45366-2	15%	

The project will be Evaluated for the following Course Outcomes

CO1: *Analyze* the impact of software engineering models over various context of software development to assess societal, health, safety, legal and cultural issues.	Total Marks	
Project Background Analysis and feasibility (needs, goal, benefits, etc.)	[5 Marks]	
Analysis the impact of societal, health, safety, legal and cultural issues	[5Marks]	
Review of existing Studies and Relevant Example	[5Marks]	
CO2: *Explain* appropriate software engineering model, project management roles and their skills in the context of professional engineering practice and solutions to complex engineering problems in a software development environment.	Total Marks	
Appropriate Process Model Selection and Argumentation with Evidence	[5Marks]	
Evidence of Argumentation regarding process model selection	[5Marks]	
Submission, Defense, Completeness, Spelling, grammar and Organization of the Project report	[5Marks]	

Student Name: KHONDOKER MD. SABIT HASAN Student ID: 21-45306-2 Contribution in Percentage (20%): <u>Contribution in the Project:</u> ▪ Project process model ▪ Use Case diagram. ▪ Functional requirements ▪ UI/UX design ▪ Test Case ▪ Work breakdown structure ▪ Effort estimation ▪ EVA _________Sabit___________ Signature of the Student
Student Name: NOKIBUL ARFIN SIAM Student ID: 21-44793-1 Contribution in Percentage (20%): <u>Contribution in the Project:</u> ▪ Project proposal (Solution to the problem) ▪ Sequence Diagram ▪ UI/UX design ▪ Functional requirements ▪ Work breakdown structure ▪ Timeline Chart ▪ Test case ______Siam_______ Signature of the Student
Student Name: ABU NASER MD. ARMAN Student ID: 21-45239-2 Contribution in Percentage (20%): <u>Contribution in the Project:</u> ▪ Project Proposal (Background to the problem) ▪ Functional requirements ▪ Activity Diagram ▪ Work breakdown structure _________Arman___________ Signature of the Student
Student Name: FAHIM RAHMAN Student ID: 21-44399-1 Contribution in Percentage (20%): <u>Contribution in the Project:</u> ▪ Functional requirements ▪ Class diagram

▪ Timeline Chart ▪ Building risk table __________Fahim____________ Signature of the Student
Student Name: MD. NAJIB HOSSAIN Student ID: 21-45366-2 Contribution in Percentage (20%): <u>Contribution in the Project:</u> ▪ Project process model ▪ Class diagram ▪ Test Case ▪ EVA ▪ Timeline Chart ▪ Building risk table __________Najib____________ Signature of the Student

Contents

1. PROJECT PROPOSAL

1.1 Background to the problem:

Agriculture is the primary source of livelihood for millions of people worldwide. Efficient irrigation practices can help ensure the long-term viability of farming and support rural communities. Farmers often rely on manual observations and subjective judgment in traditional irrigation methods to determine when and how much to water their crops. This method can be time-consuming and labor-intensive and may result in inconsistent irrigation practices. Furthermore, the increasing demand for food production and the growing scarcity of water resources make adopting more efficient irrigation techniques imperative.

1.2 Solution to the problem:

An automated Irrigation System is a solution aimed at improving the efficiency and productivity of agriculture by automating the irrigation process. This system utilizes soil moisture and weather data to determine the optimal water required for each crop. This helps farmers conserve water resources and avoid over-irrigation, leading to soil degradation and reduced crop yields. The Automated Irrigation System provides solutions to these problems using sensors and weather monitoring technology to collect data on soil moisture levels and weather conditions. This information will determine the optimal water required for each crop and automate the irrigation process. The system can also be programmed to consider crop type, soil type, and local weather patterns to optimize irrigation schedules. By automating the irrigation process, the Automated Irrigation System helps farmers to improve their yields, conserve water resources, and reduce labor costs. Some auto-irrigation software exists, but they don't check weather conditions. However, our software will take the weather information and make decisions based on the information that shows how much water is needed. Also, our study has utilized that the existing studies of irrigation process are too costly. Additionally, our study has focused on developing a scalable and purchasable cost solution for automated irrigation, making it accessible for small-scale farmers and agriculture firms. Furthermore, our study has emphasized incorporating real-time data analysis and visualization to help farmers make informed decisions regarding irrigation. This makes our study a significant contribution to the field of automated irrigation systems for irrigation. The target users for the "Automated Irrigation System that automates and controls the irrigation of crops based on soil moisture levels and weather conditions" are farmers and agricultural firms. This system is a step towards sustainable agriculture and helps ensure the long-term viability of farming.

2. FUNCTIONAL REQUIREMENTS

1. Sing up

1.1 User registration form will allow users to create an account on the app by entering their name, username, email, and password.
1.2 Email verification process to confirm the validity of the email address.
1.3 Username & Password will be used to log into the site.
1.4 Option to log in using a social media account (e.g., Facebook, Google).

2. Software Login

2.1 The software will allow users to log in with their username and password.
2.2 The login credentials (username and password) will be verified with database records.
2.3 The user account's home page will be displayed if the login is successful.
2.4 If the username and/or password has been inserted wrong, the random verification code, the system will generate and send the user's email address to retry login.
2.5 If the number of login attempt exceed its limit (3 times), the system shall block the user account login for one hour [optional function]

3. Weather forecast

3.1 The app will display the current weather conditions for the user's preferred location, including temperature, humidity, the possibility of rain, and wind speed. The information will be updated in real-time to provide the most accurate and up-to-date information.
3.2 Option to view the 7-day forecast, including high and low temperatures and weather conditions. This will help users plan for the week ahead and prepare for adverse weather conditions.
3.3 Ability to change the location to view weather forecasts for different cities.

4. Settings

4.1 Location: Ability to set a preferred location to view weather forecasts.
4.2 Language: Option to select a preferred language for the app.
4.3 Profile: Ability to view and edit personal information, such as name and email address. This will allow them to keep their knowledge up to date.
4.4 Password: Ability to change the old password.

5. Community Feedback

5.1 Users can provide feedback on the app, including suggestions for improvement.
5.2 Option for users to rate the app and leave reviews.
5.3 Developers can respond to user feedback and address any concerns. This will help build trust with users and demonstrate the developers' commitment to improving the app.

6. Check soil moisture

6.1 The app will allow users to check the moisture levels in their soil. This is useful for farmers to monitor the health of their crops and ensure they are getting the water they need.
6.2 The app will allow users to select their preferred soil type to provide the most accurate readings. This will consider the different moisture requirements of different kinds of soil.
6.3 Ability to view historical data on soil moisture levels.

7. Notifications

7.1 Option to receive notifications for important events, such as changes in weather alerts or reminders to check soil moisture.
7.2 Gives notification for agriculture articles, news, etc.
7.3 Gives notification if helpline or support center responds.

8. Helpline

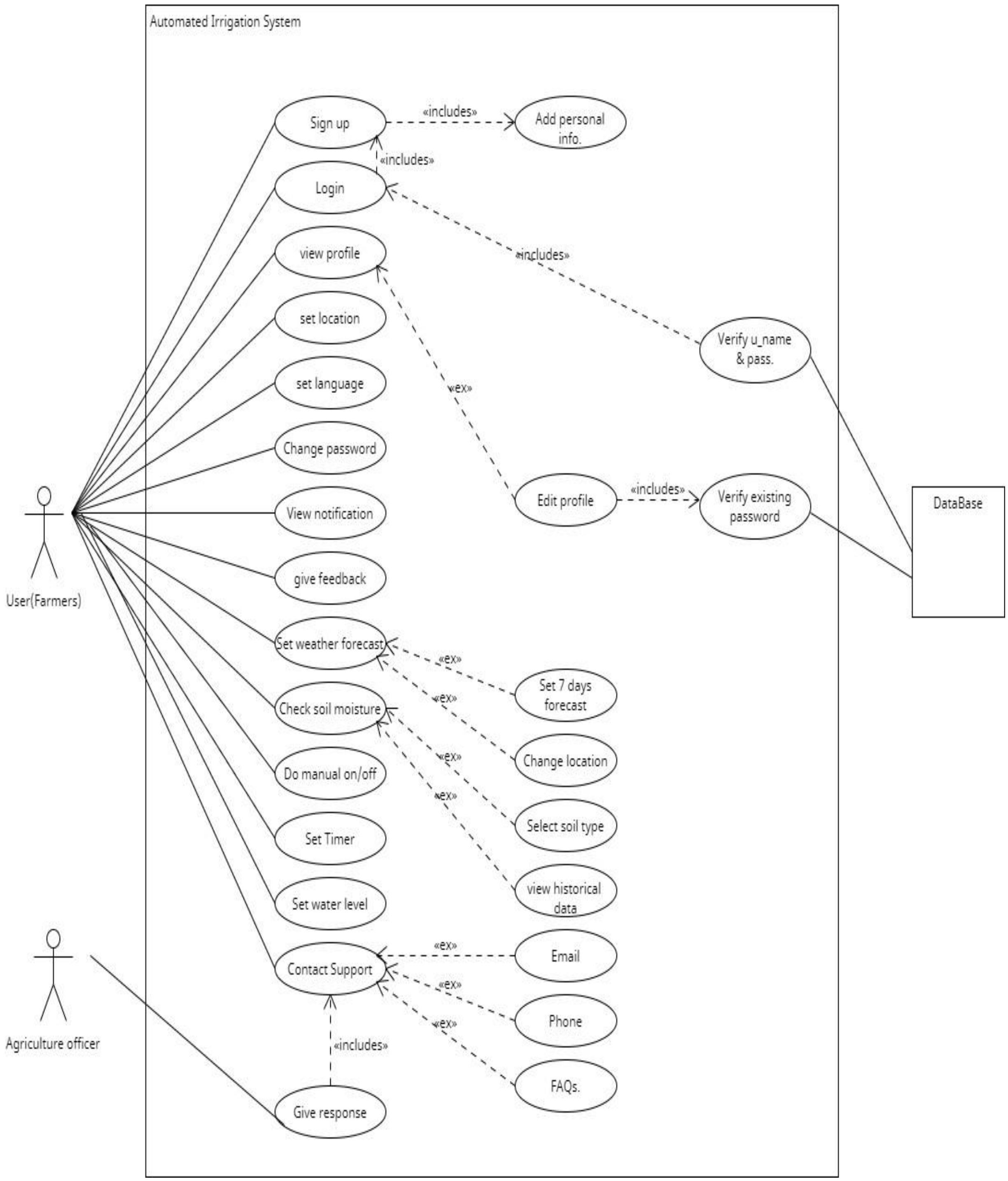
8.1 Access a helpline or support center for assistance with any issues or questions.
8.2 Option to contact support via email or phone.
8.3 FAQs: The app will have a section with frequently asked questions and answers to help users find the information they need quickly and easily.

9. Irrigation system setting

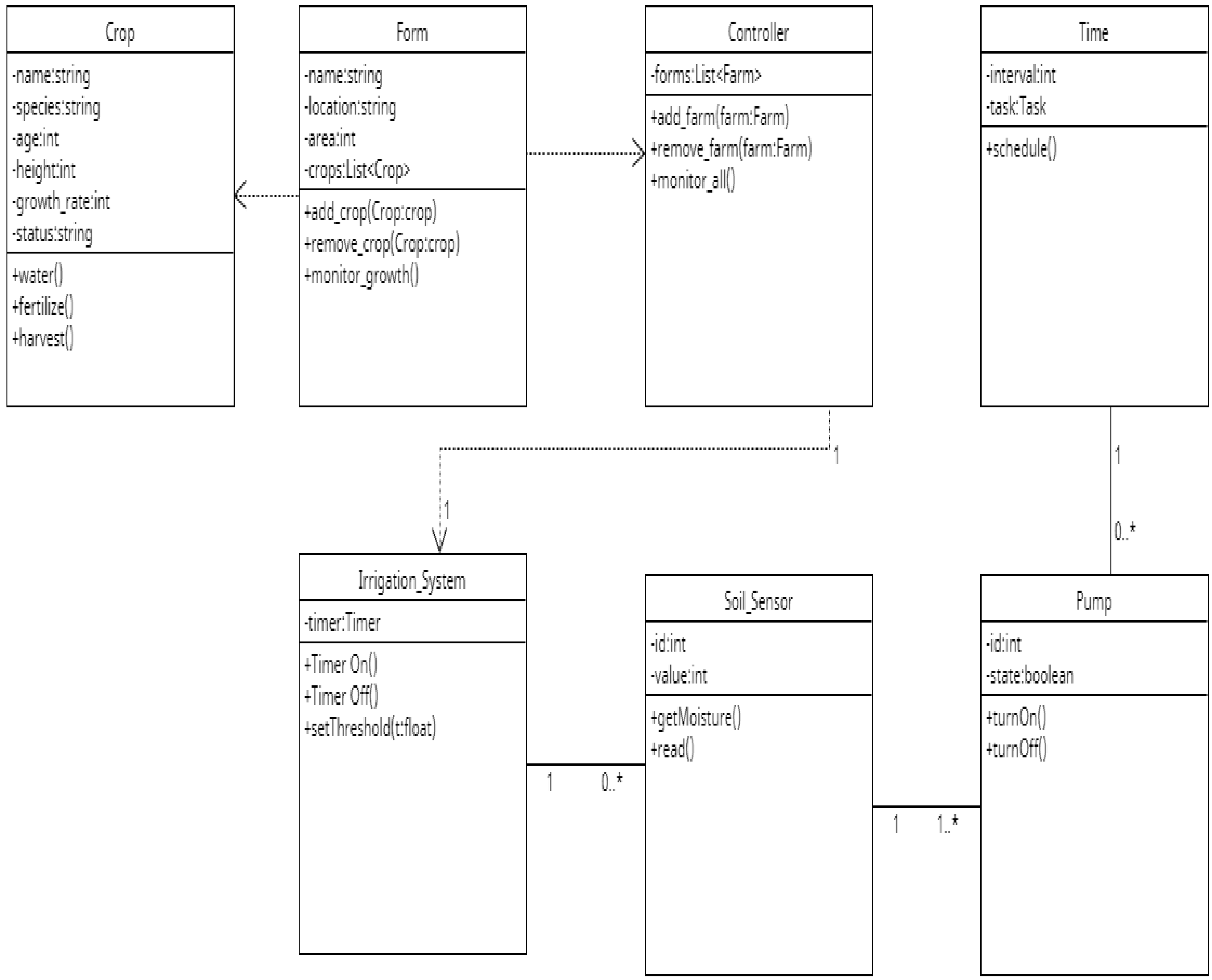
9.1 Auto On/Off: The app will be able to turn the irrigation system on and off automatically, based on pre-defined settings such as weather conditions or soil moisture levels.
9.2 Manual On/Off: The user can manually operate (On/Off) the irrigation system.
9.3 Set Timer: Ability to set a timer for the irrigation system to turn on and off.
9.4 Set Water Level: Option to set a preferred water level for the irrigation system.

3. DIAGRAMS

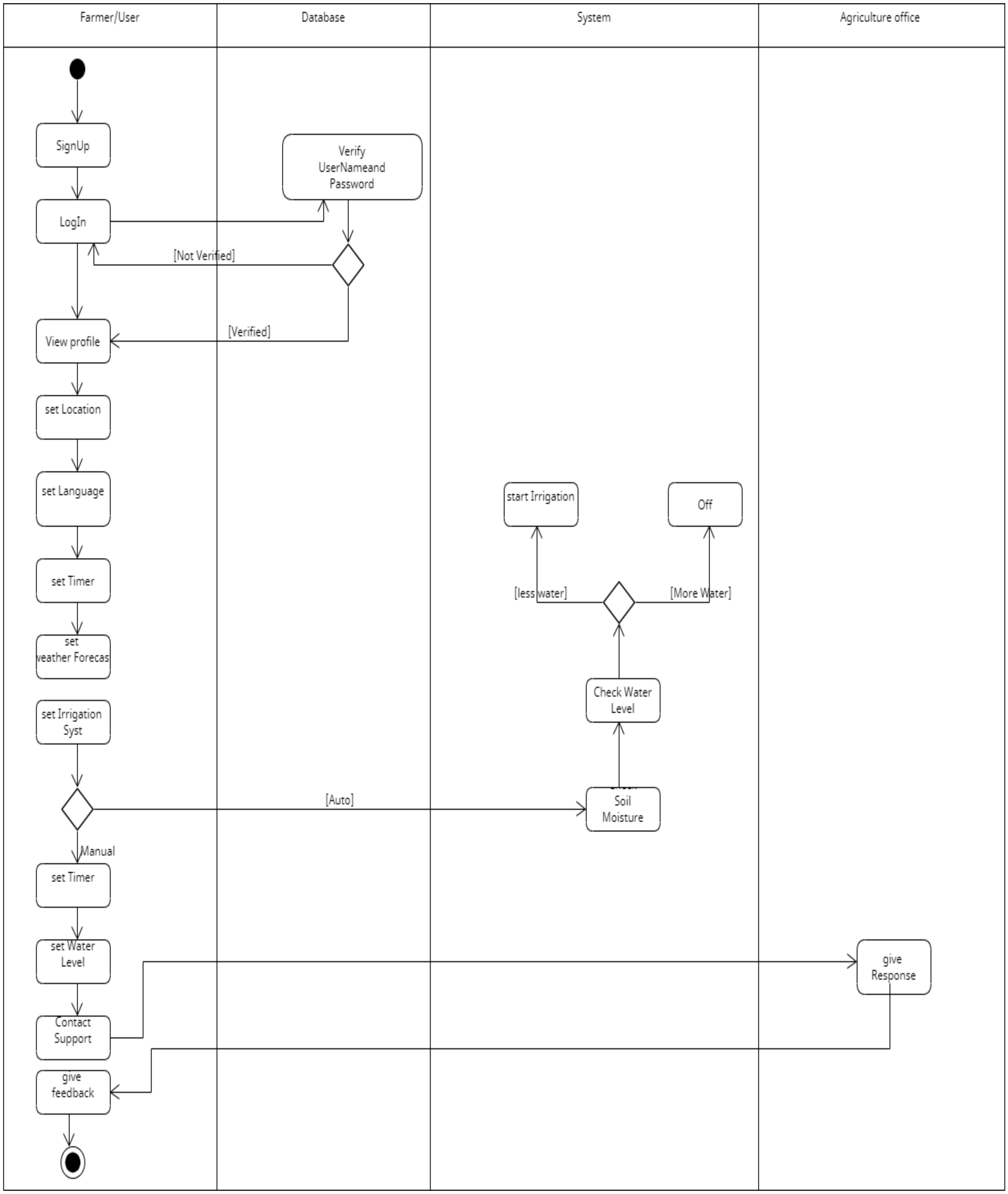
3.1: Use Case Diagram



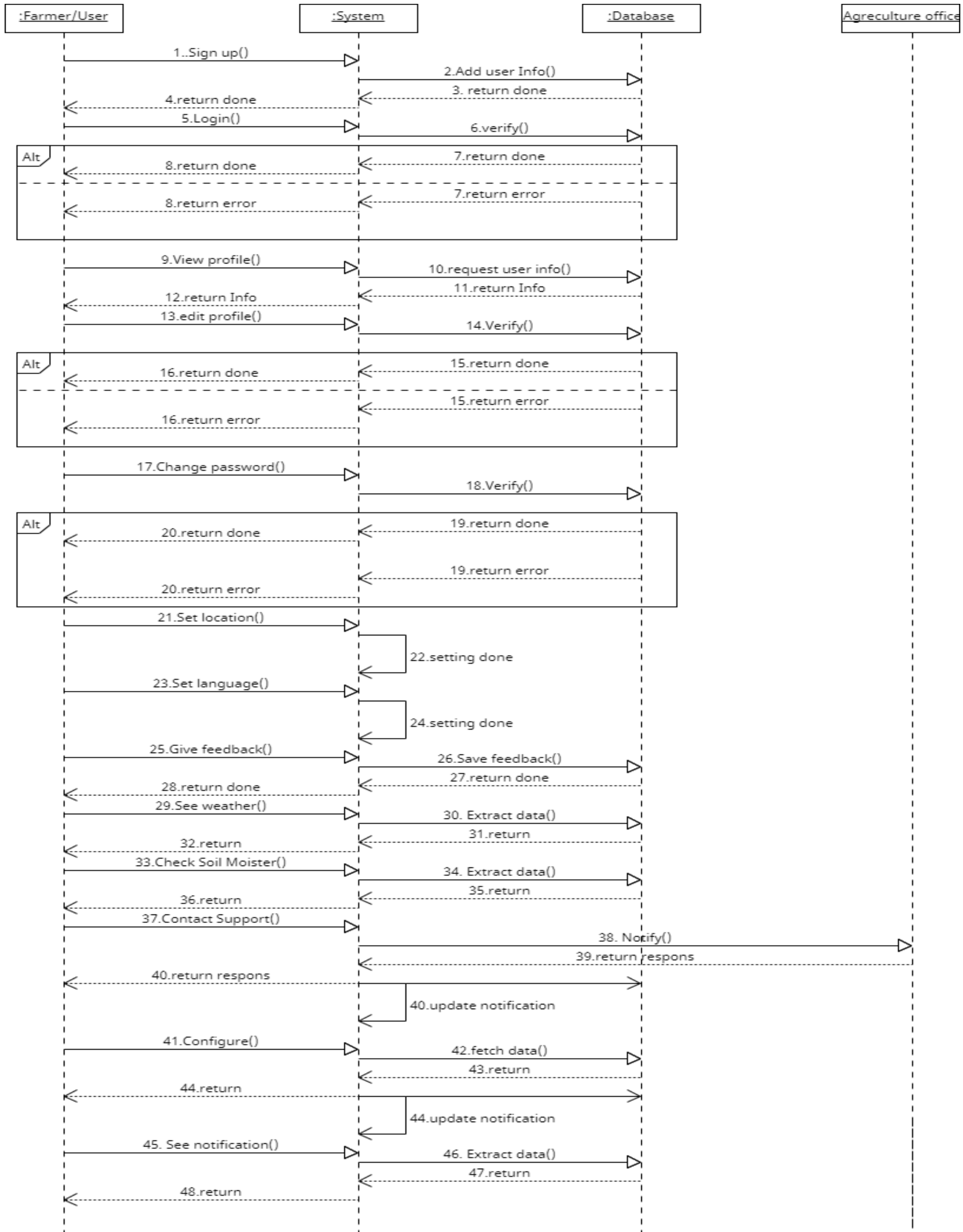
3.2: Class Diagram



3.3: Activity Diagram



3.4 Sequence Diagram



4. SOFTWARE DEVELOPMENT LIFE CYCLE

4.1 Process Model

For an Automated Irrigation System that automates and controls the irrigation of crops based on soil moisture levels and weather conditions, the software engineering model that is best suited is the Agile Model.

The Agile Model is an iterative and flexible software development approach, allowing for changes to be made quickly and efficiently as requirements and specifications change. This model is particularly well-suited for projects where requirements are likely to change frequently, such as in an agricultural system that relies on real-time weather data and soil moisture levels, which is the same as our project. By adopting an Agile development process, the Automated Irrigation System can be built flexibly and responsively, allowing for changes to be made quickly and efficiently as the system evolves. This will ensure that the system remains effective and efficient, providing optimal irrigation to crops based on real-time data and ultimately improving crop yields and reducing water waste.

But why not a plan-driven software development process? Because it is a traditional approach that emphasizes creating a comprehensive plan before starting development work. This process will not be well-suited for developing an automated irrigation system for several reasons:

1. Our automated irrigation system needs to respond to changing environmental conditions in real time, such as changes in weather, soil moisture levels, and plant water requirements. This means that the system must adapt to these changes quickly and dynamically, which may not be possible with a plan-driven approach that focuses on following a predetermined plan.
2. Our automated irrigation system involves multiple sensors and controllers that must work together seamlessly. Such a complex system may be challenging to plan for in advance, and changes or updates to the plan may be needed frequently during development.

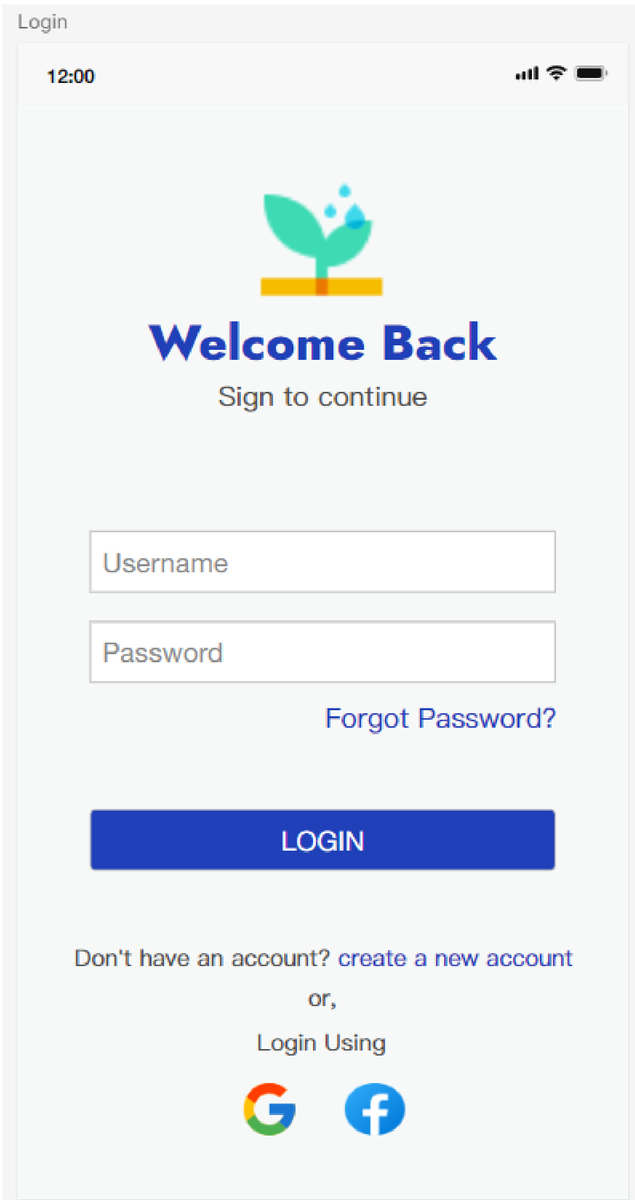
Therefore, a more flexible and adaptive approach, agile software development, will be more suitable for developing an automated irrigation system. But there are several Agile models, each with specific characteristics and methodologies. However, for an Automated Irrigation System that automates and controls the irrigation of crops based on soil moisture levels and weather conditions, the Scrum model would be a good fit.

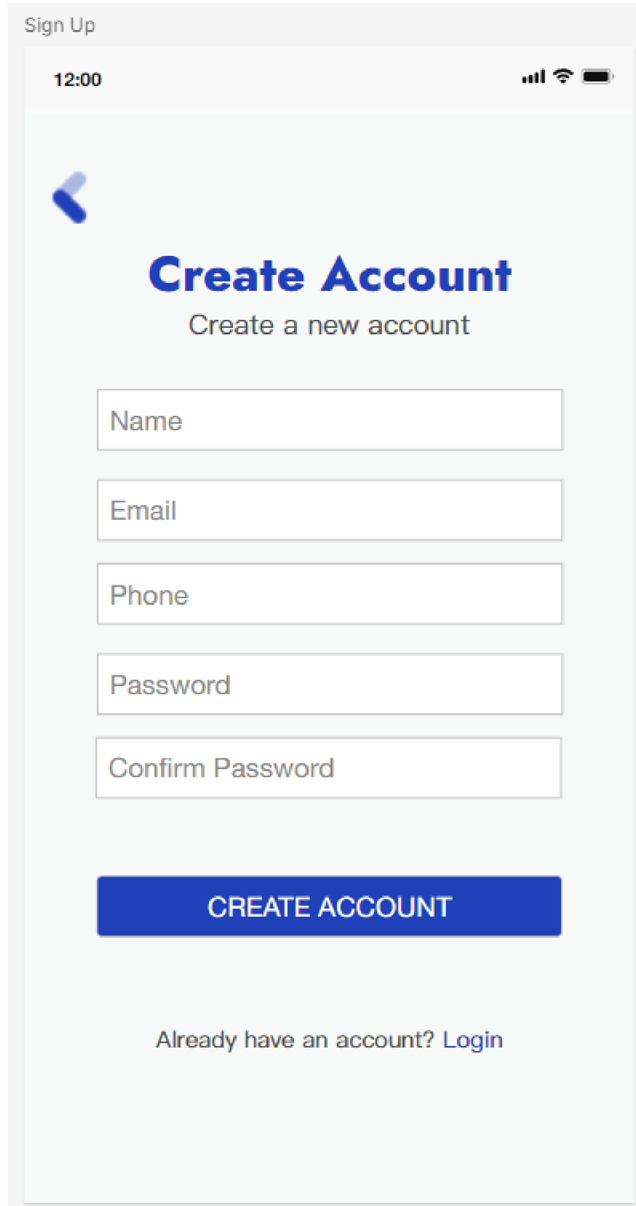
Scrum is an Agile model that focuses on delivering working software in short iterations called sprints. The development process is broken down into smaller, manageable tasks called user stories. Each sprint involves selecting a set of user stories to work on, completing them, and delivering a working product increment.

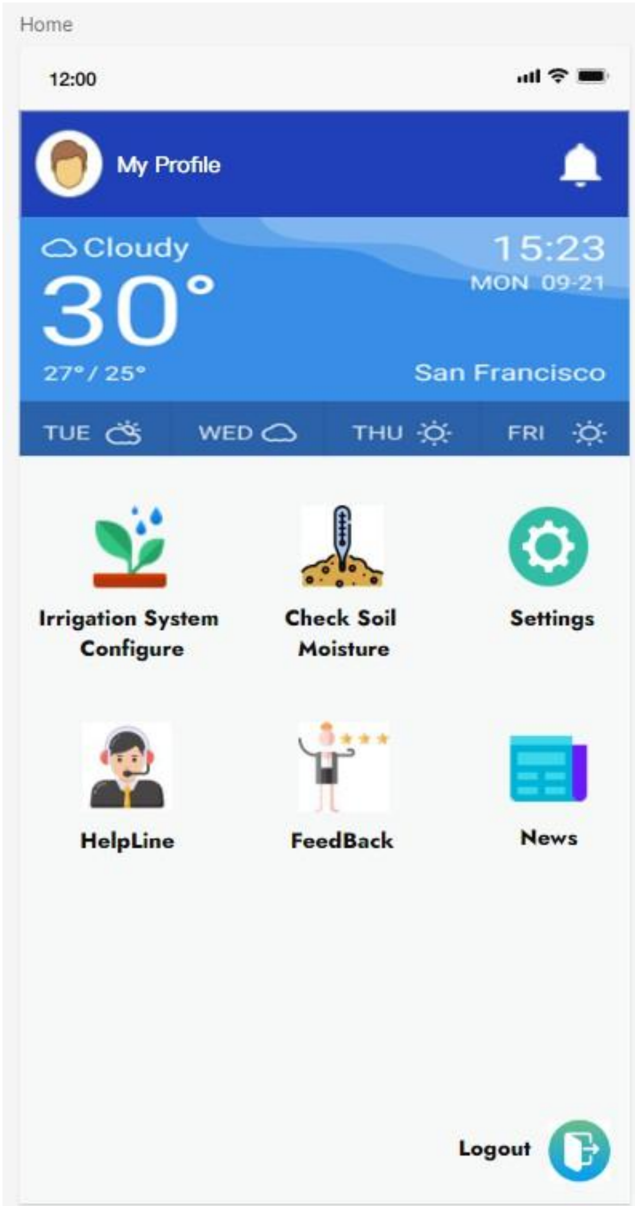
In the case of an Automated Irrigation System, each sprint could focus on adding new features or improving existing functionality, such as integrating new weather sensors or improving the system's ability to analyze soil moisture data. Scrum also emphasizes frequent communication and collaboration between the development team and stakeholders, with daily stand-up meetings, sprint reviews, and retrospectives. This ensures that the system remains aligned with the needs of the business and that any issues or concerns are addressed quickly.

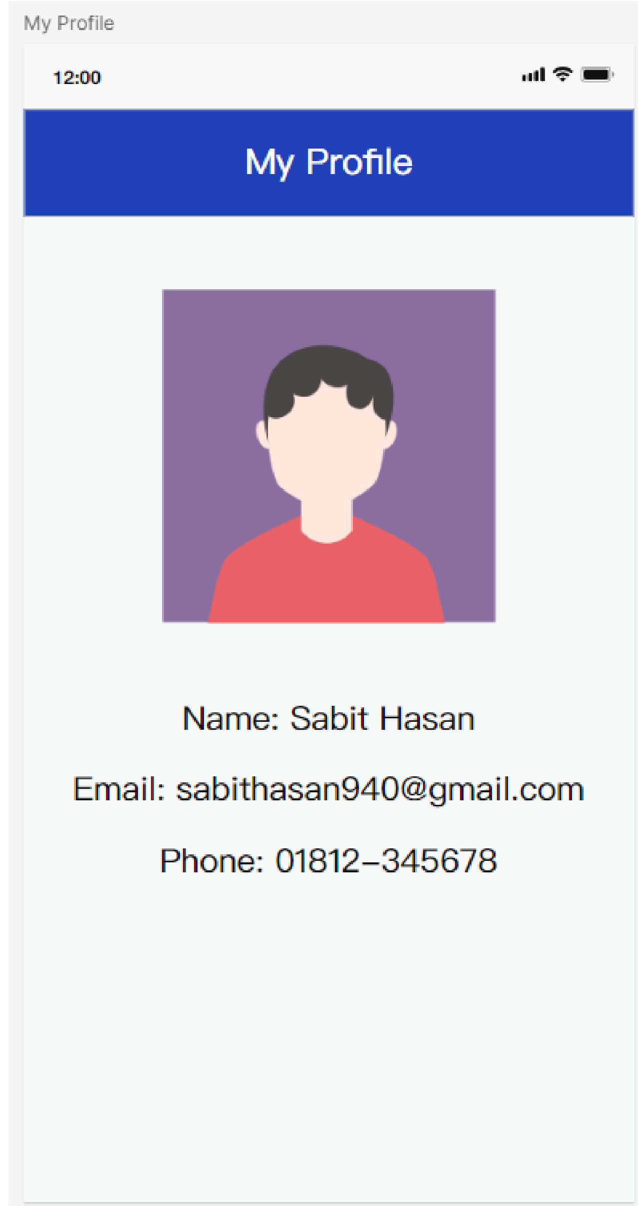
Overall, the Scrum model would be a good fit for our Automated Irrigation System as it provides a flexible and responsive development process that can adapt to changing requirements and priorities while ensuring that the system is delivered in working increments that provide real value to the business.

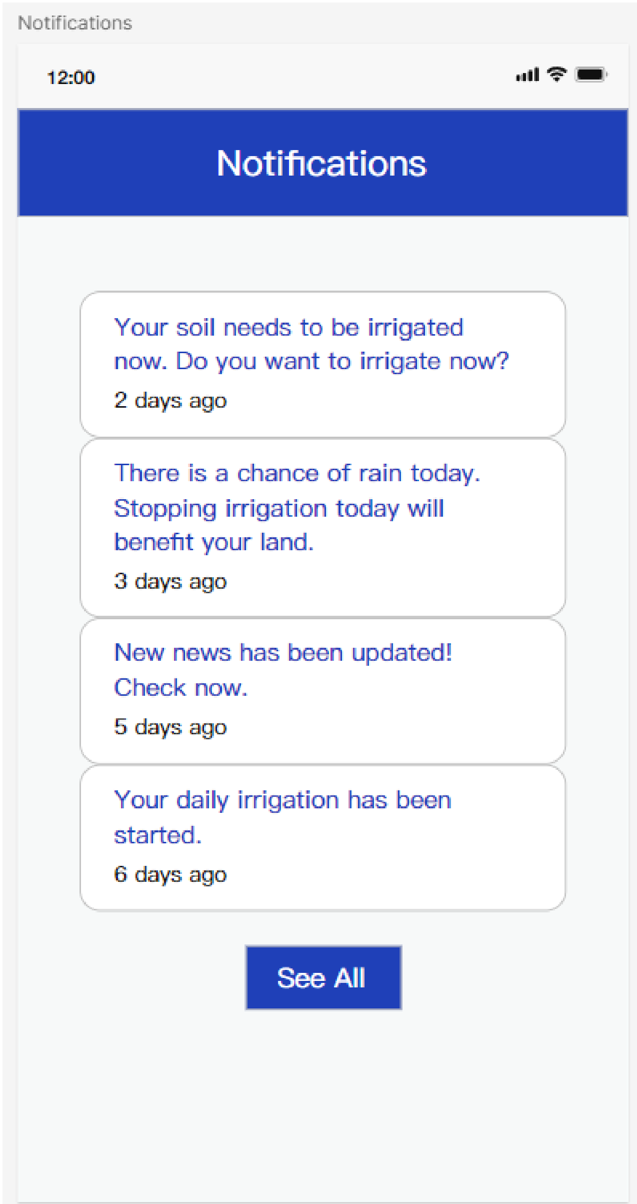
5. USER INTERFACE AND EXPERIENCE (UI/UX)



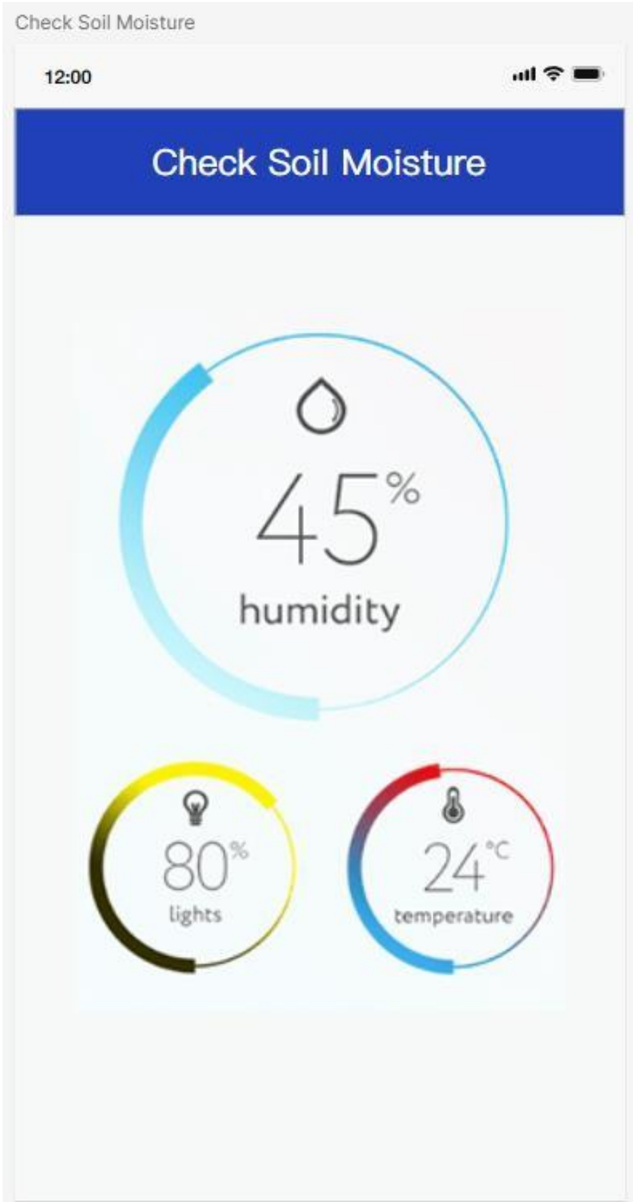


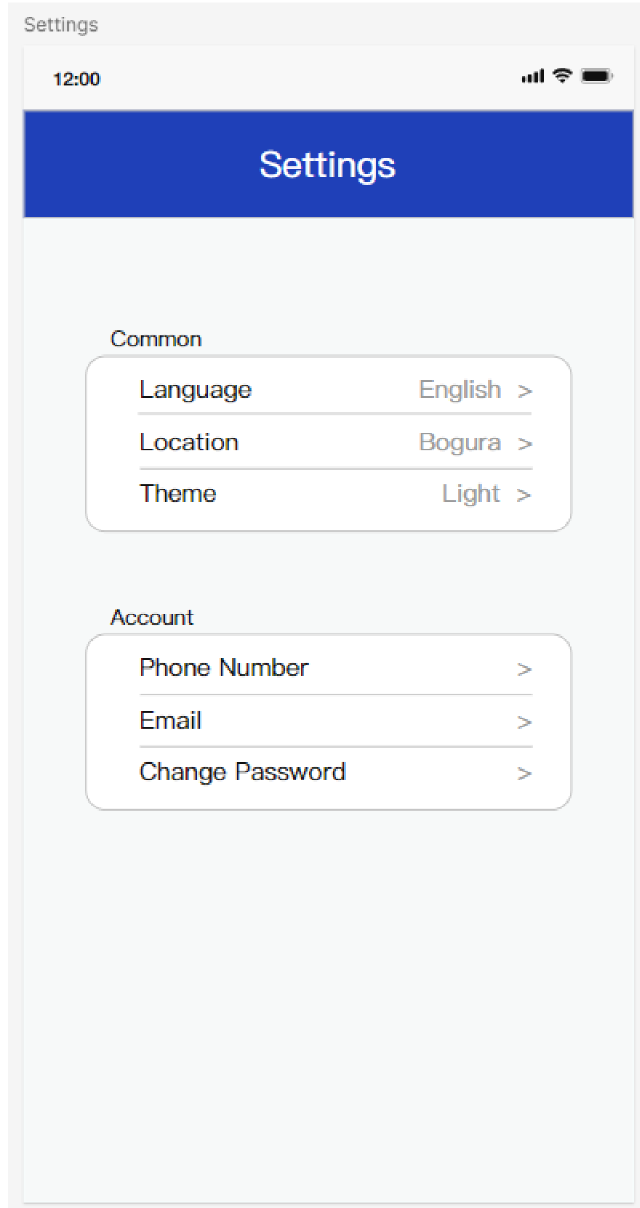

Home
12:00
My Profile
Cloudy
15:23
MON 09-21
30°
27° / 25°
San Francisco
TUE
WED
THU
FRI
Irrigation System Configure
Check Soil Moisture
Settings
HelpLine
FeedBack
News
Logout


My Profile
12:00
My Profile
Name: Sabit Hasan
Email: sabithasan940@gmail.com
Phone: 01812-345678

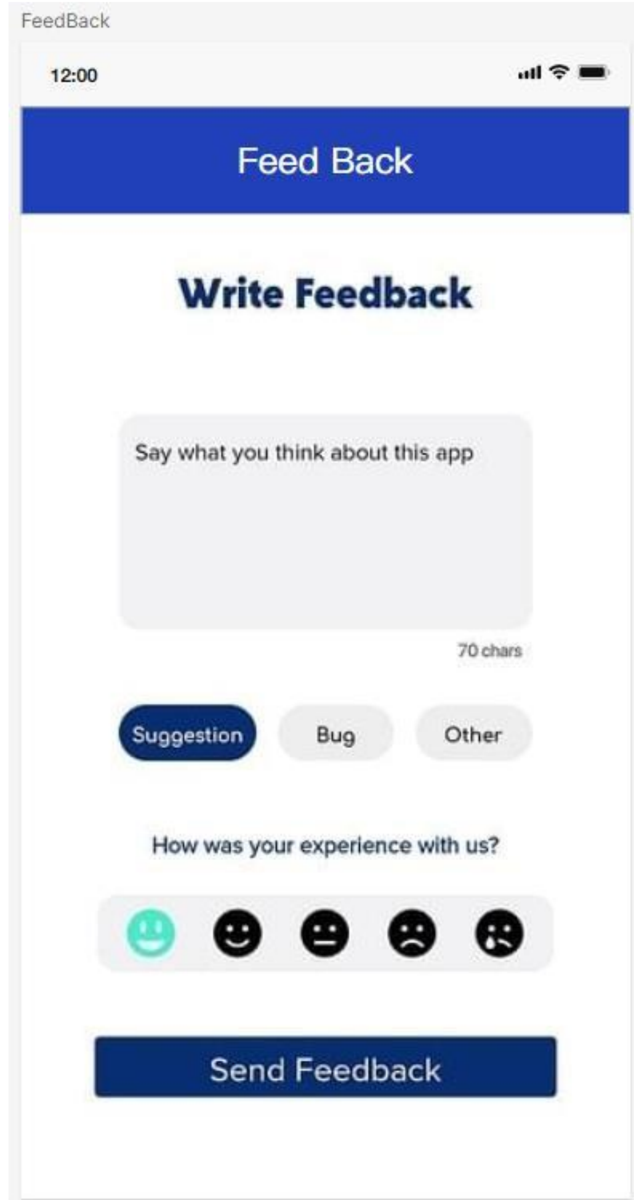

Notifications
12:00
Notifications
Your soil needs to be irrigated now. Do you want to irrigate now?
2 days ago
There is a chance of rain today. Stopping irrigation today will benefit your land.
3 days ago
New news has been updated! Check now.
5 days ago
Your daily irrigation has been started.
6 days ago
See All


Irrigation System Configure
12:00
Irrigation System Configure
ON/OFF
Schedule
Timer
Loop Timer

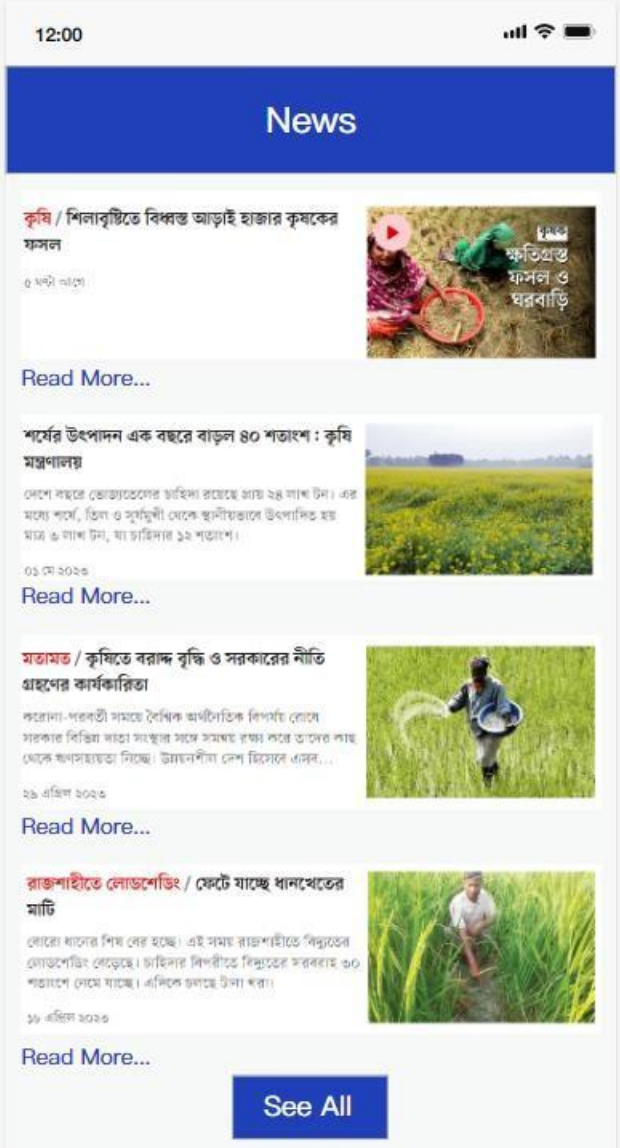

Check Soil Moisture
12:00
Check Soil Moisture
45%
humidity
80%
lights
24°C
temperature


Settings
12:00
Settings
Common
Language English >
Location Bogura >
Theme Light >
Account
Phone Number >
Email >
Change Password >


HelpLine
12:00
Help Line
Get help
Call Us
Start Chat


FeedBack
12:00
Feed Back
Write Feedback
Say what you think about this app
70 chars
Suggestion
Bug
Other
How was your experience with us?
Send Feedback

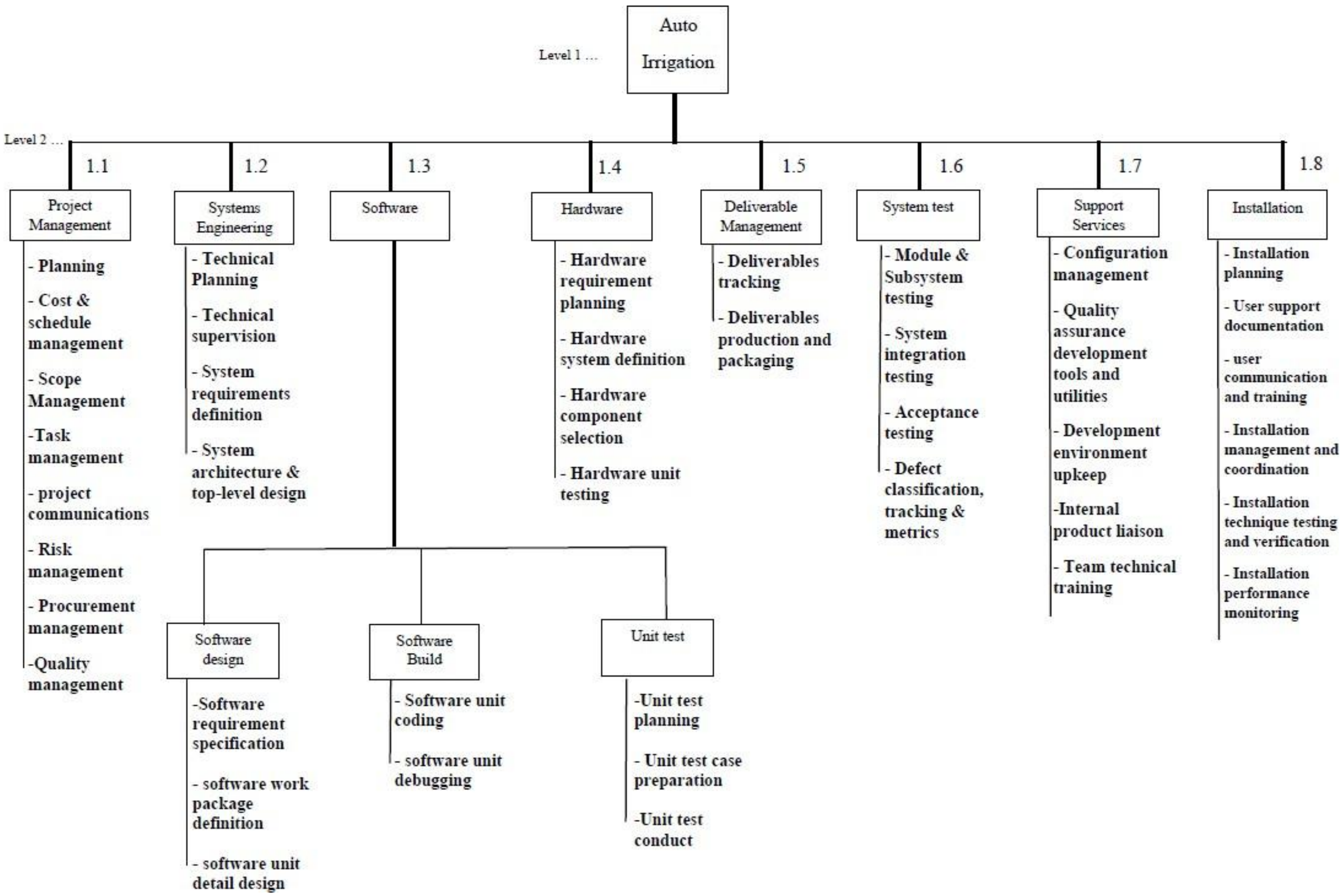
News


12:00
News
কৃষি / শিলাবৃষ্টিতে বিধ্বস্ত আড়াই হাজার কৃষকের ফসল
ক্ষতিগ্রস্ত ফসল ও ঘরবাড়ি
Read More...
শর্ষের উৎপাদন এক বছরে বাড়ল ৪০ শতাংশ : কৃষি মন্ত্রণালয়
Read More...
মতামত / কৃষিতে বরাদ্দ বৃদ্ধি ও সরকারের নীতি গ্রহণের কার্যকারিতা
Read More...
রাজশাহীতে লোডশেডিং / ফেটে যাচ্ছে ধানখেতের মাটি
Read More...
See All

6. TEST CASES/TEST ITEMS

***** View Excel file for Test Cases *****

7. WORK BREAKDOWN STRUCTURE (WBS)



8. EFFORT ESTIMATION

Constructive Cost Model:

SLOC = 8000
P = 1.20 (Embedded)
T = 0.32

Effort = PM = Coefficient<Effort Factor>*(SLOC/1000) ^P
= 3.6 * (8000/1000) ^1.20
= 43.65 months

Development time = DM = 2.50*(PM)^T
= 2.50 * (43.65) ^0.32
= 8.37 months

Required number of people = ST = PM/DM
= 43.65/8.37
= 5.21
= 5

EVA:

Task	Planned Effort	Actual Effort
1	12.0	12.5
2	15.0	11.0
3	13.0	17.0
4	8.0	9.5
5	9.5	9.0
6	18.0	19.0
7	10.0	10.0
8	4.0	4.5
9	12.0	10.0
10	6.0	6.5
11	5.0	-
12	14.0	-
13	16.0	-
14	6.0	-
15	8.0	-

When we were asked to do the earned value analysis, 10 tasks were completed. However, the project schedule indicates that 15 tasks should have been completed.

Effort Estimated = 1310 Person Day

BAC = 1310.00

BCWP = 107.5

BCWS = 156.5

ACWP = 109

SPI = BCWP/BCWS = 107.5/156.5 = 0.6869

SV = BCWP – BCWS = 107.5 – 156.5 = -49 person-day

CPI = BCWP/ACWP = 107.5/109 = 0.986

CV = BCWP – ACWP = 107.5 – 109 = -1.5 person-day

% schedule for completion = BCWS/BAC = 156.5/1310 = 11.947%

[% of work schedule to be done at this time]

% complete = BCWP/BAC = 107.5/1310 = 8.206%

[% of work completed at this time]

9. TIMELINE CHARTS

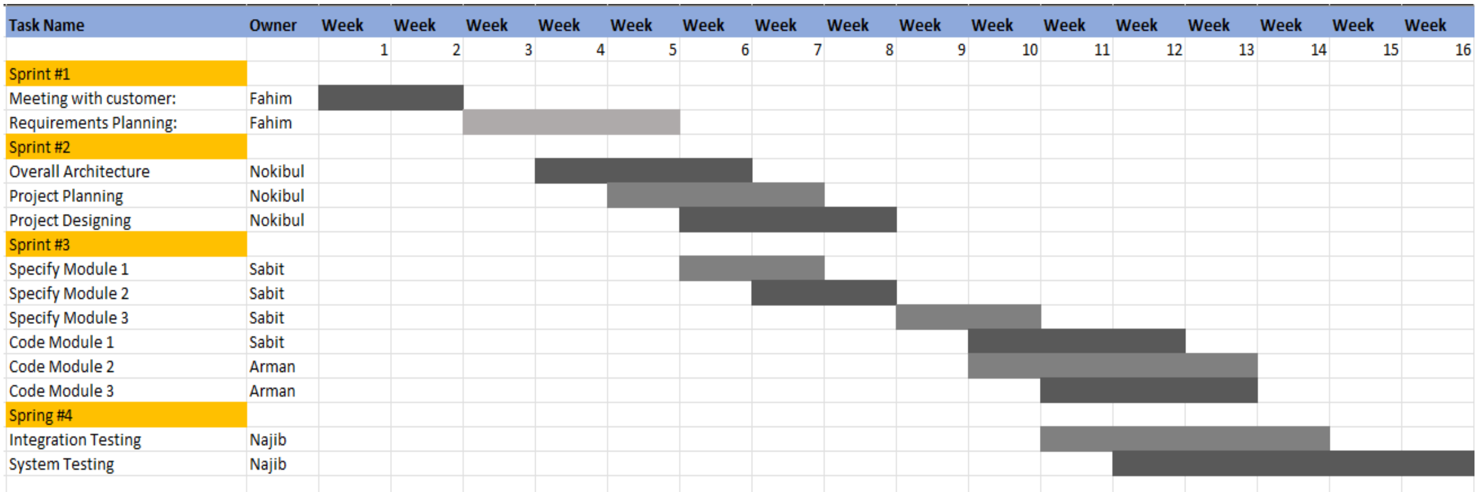
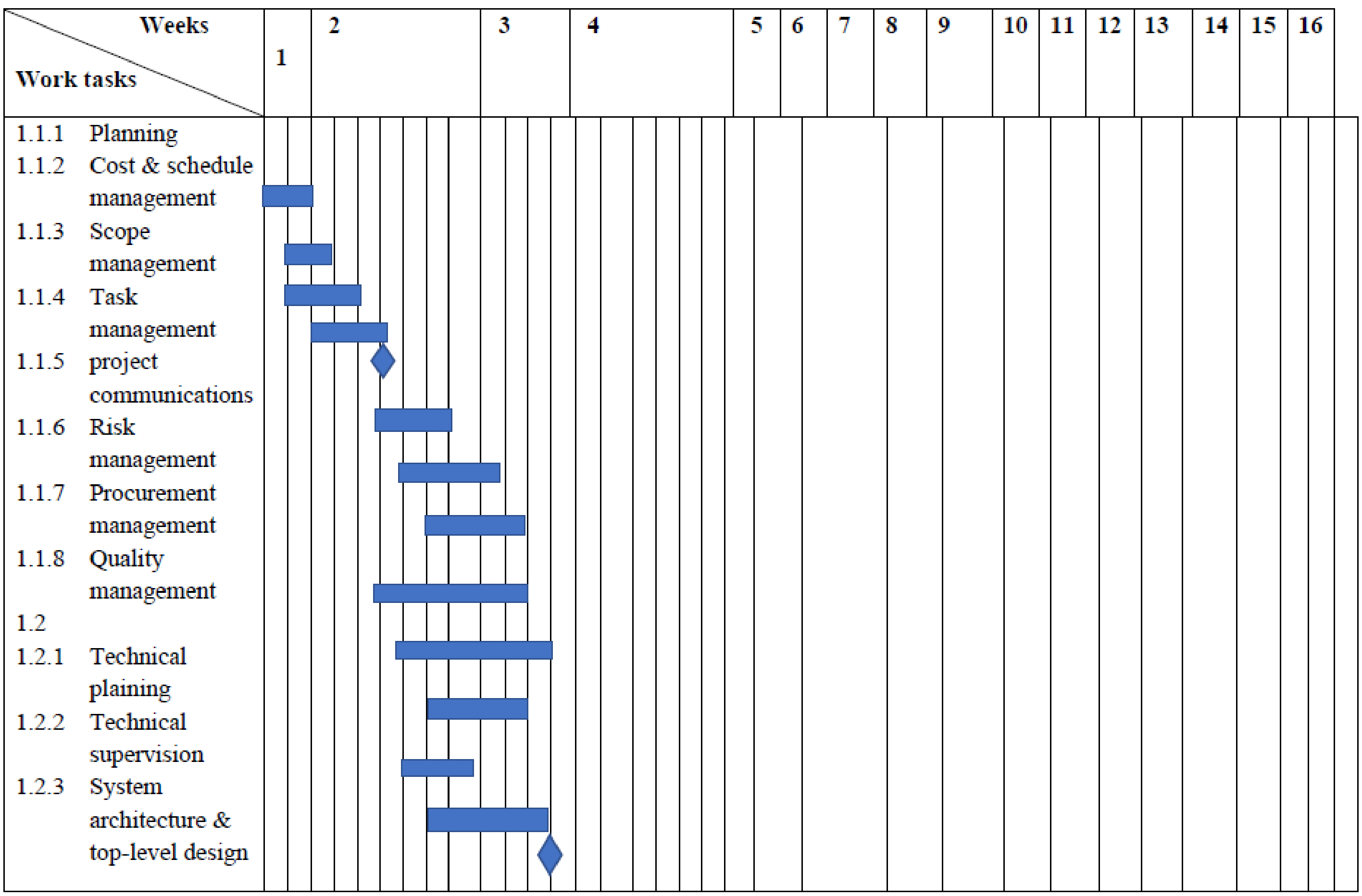
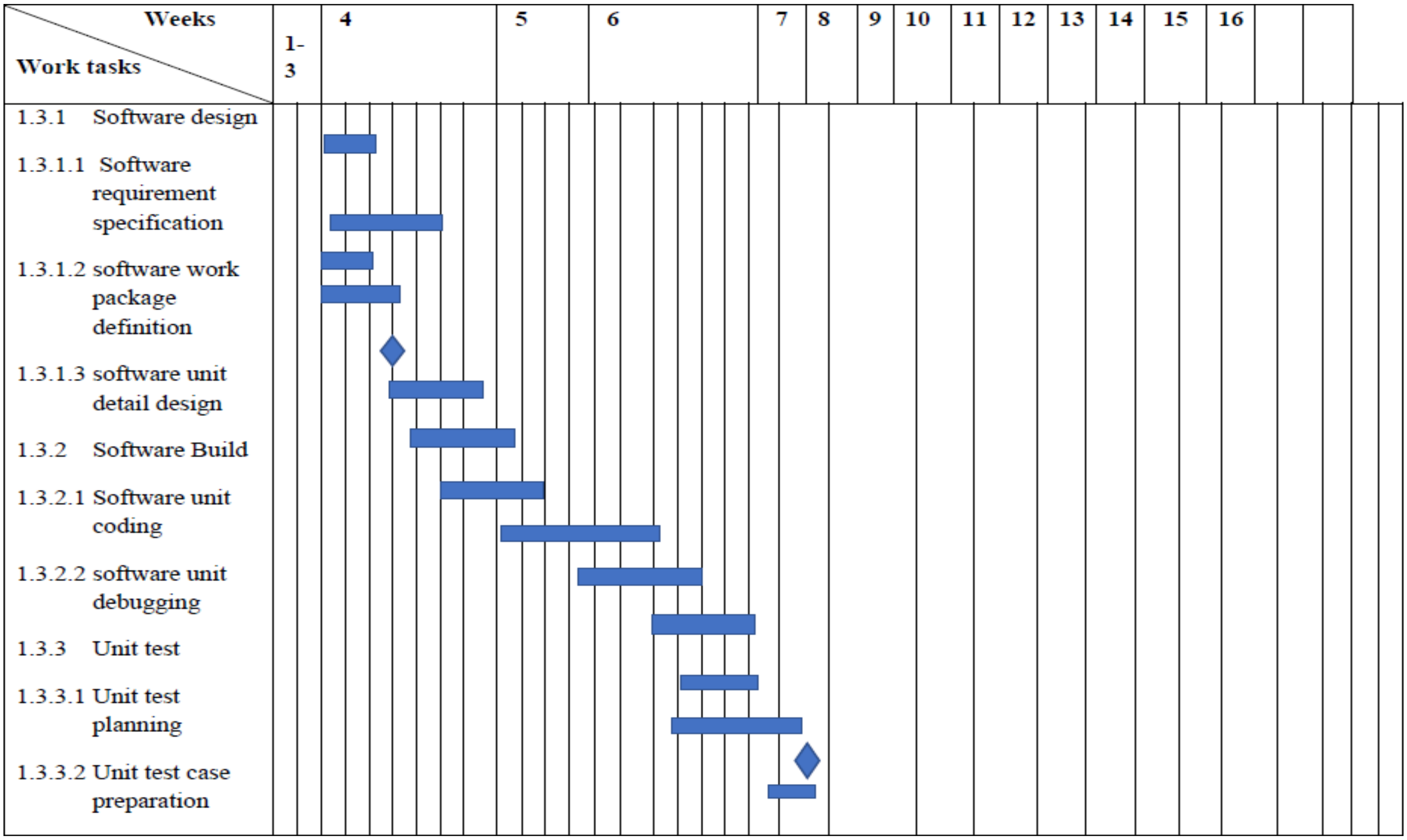
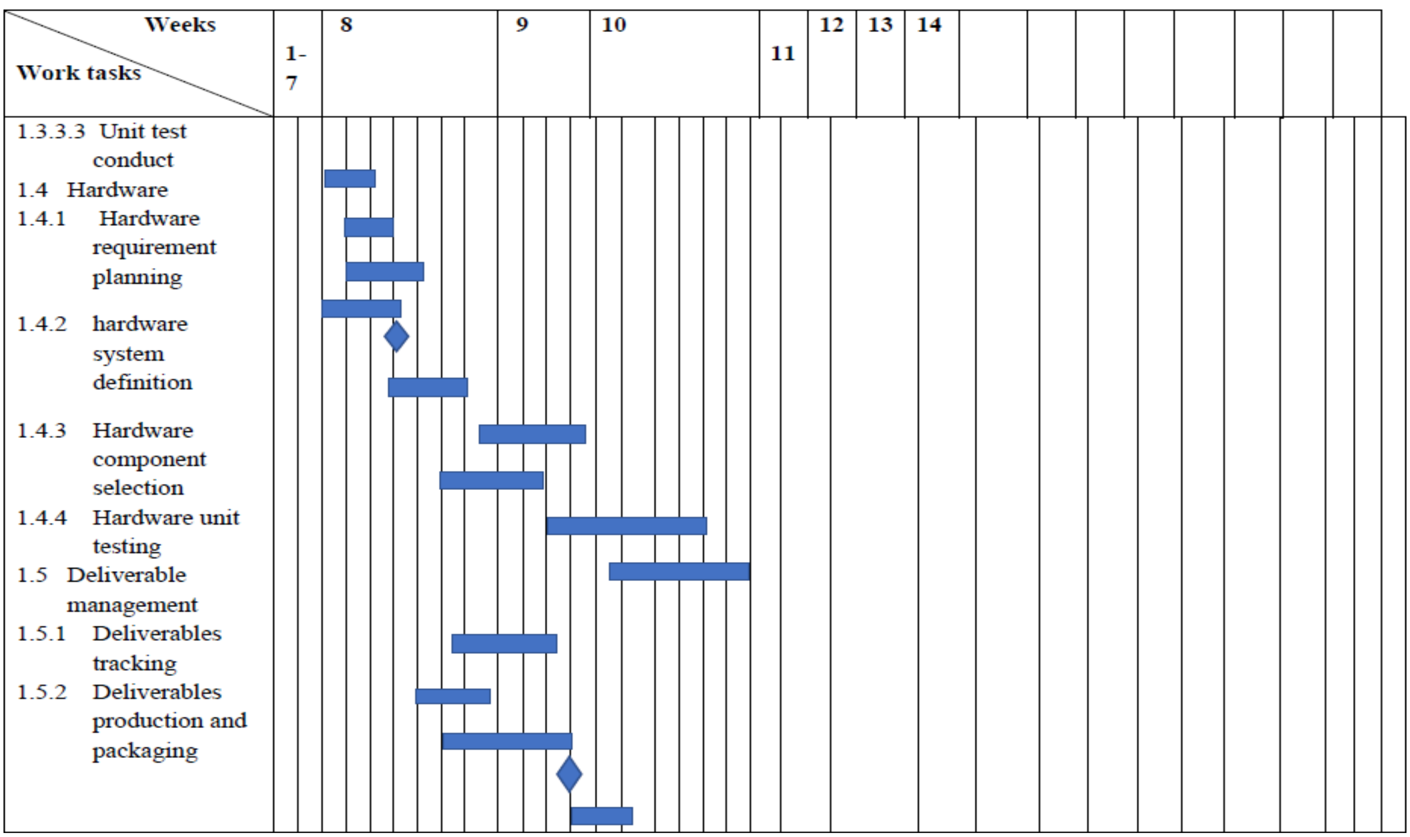
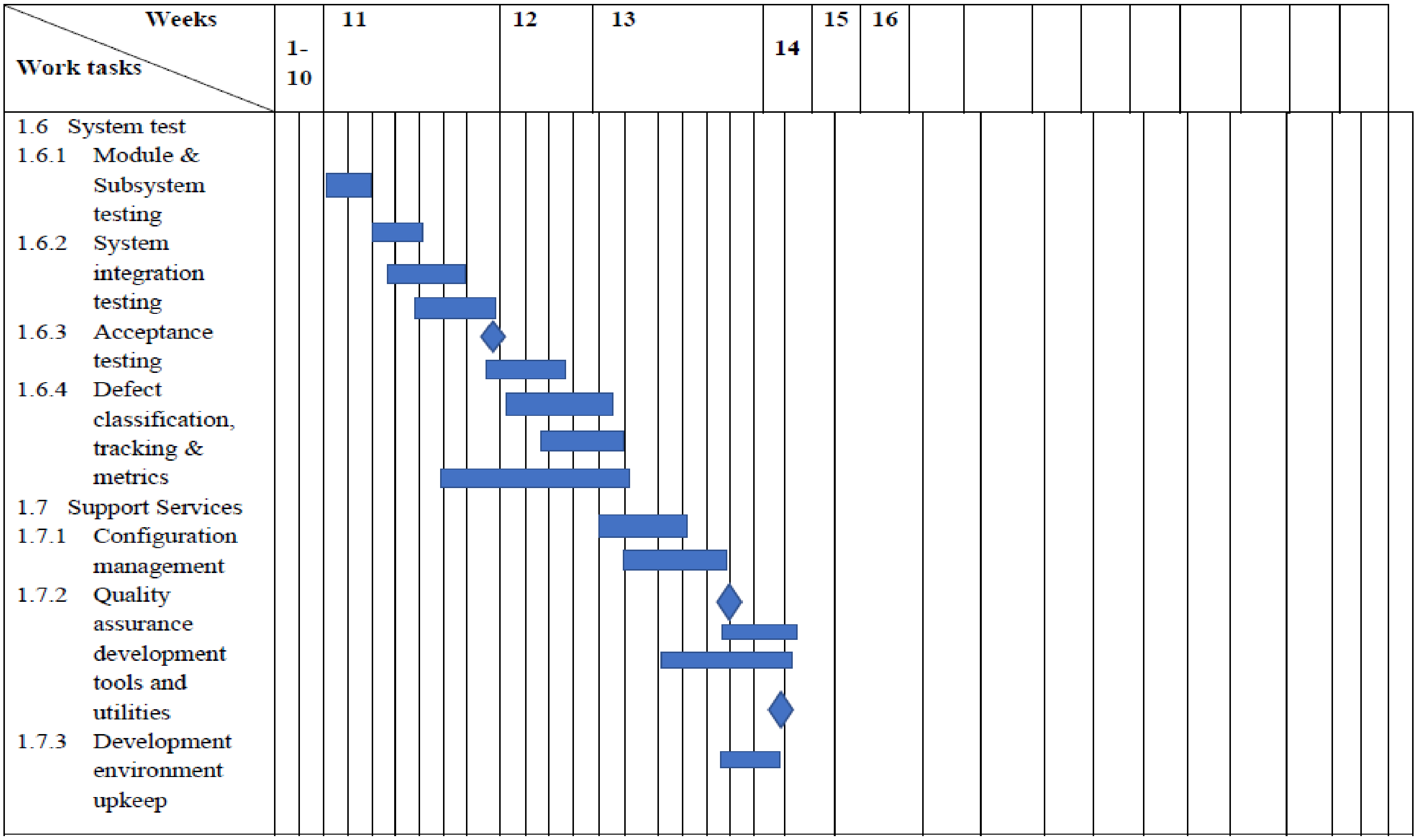


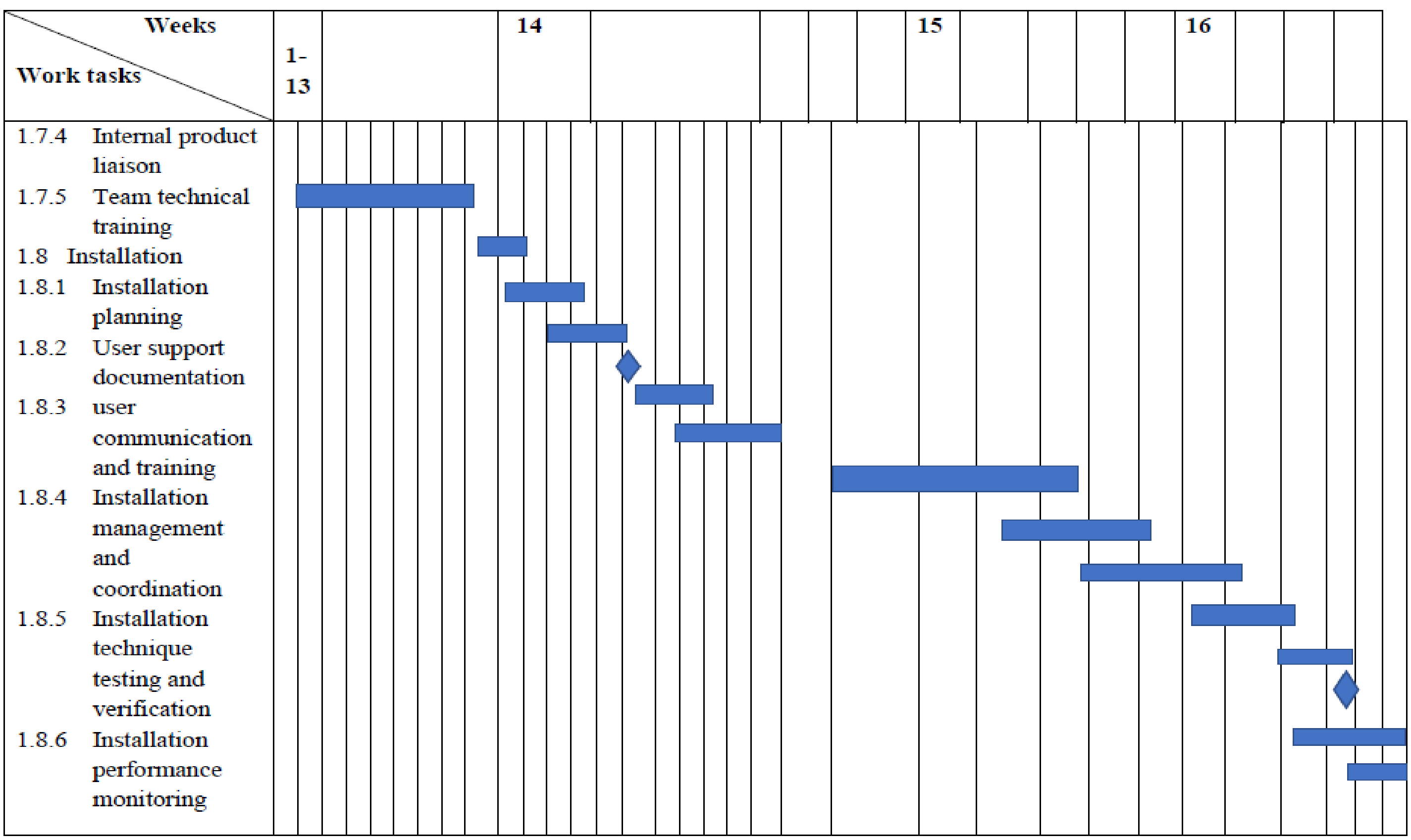
Fig: Overall project plan as a bar chart




Weeks
Work tasks
1-3
4
5
6
7
8
9
10
11
12
13
14
15
16
1.3.1 Software design
1.3.1.1 Software requirement specification
1.3.1.2 software work package definition
1.3.1.3 software unit detail design
1.3.2 Software Build
1.3.2.1 Software unit coding
1.3.2.2 software unit debugging
1.3.3 Unit test
1.3.3.1 Unit test planning
1.3.3.2 Unit test case preparation


Weeks
Work tasks
1-7
8
9
10
11
12
13
14
1.3.3.3 Unit test conduct
1.4 Hardware
1.4.1 Hardware requirement planning
1.4.2 hardware system definition
1.4.3 Hardware component selection
1.4.4 Hardware unit testing
1.5 Deliverable management
1.5.1 Deliverables tracking
1.5.2 Deliverables production and packaging


Weeks
Work tasks
1-10
11
12
13
14
15
16
1.6 System test
1.6.1 Module & Subsystem testing
1.6.2 System integration testing
1.6.3 Acceptance testing
1.6.4 Defect classification, tracking & metrics
1.7 Support Services
1.7.1 Configuration management
1.7.2 Quality assurance development tools and utilities
1.7.3 Development environment upkeep


Weeks
Work tasks
1-13
14
15
16
1.7.4 Internal product liaison
1.7.5 Team technical training
1.8 Installation
1.8.1 Installation planning
1.8.2 User support documentation
1.8.3 user communication and training
1.8.4 Installation management and coordination
1.8.5 Installation technique testing and verification
1.8.6 Installation performance monitoring

10. BUILDING RISK TABLE

Risks	Category	Probability	Impact
Size estimate may be significantly low	PS	60%	2
Larger number of users than planned	PS	30%	3
Less reuse than planned	PS	70%	2
End-users resist system	BU	40%	3
Delivery deadline will be tightened	BU	50%	2
Funding will be lost	CU	40%	1
Customer will change requirements	PS	80%	2
Technology will not meet expectations	TE	30%	1
Lack of training on tools	DE	80%	3
Staff inexperienced	ST	30%	2
Staff turnover will be high	ST	60%	2
Staff fall sick	ST	30%	2
Wrong user interface	CU	40%	2
Wrong software functions	PR	30%	1
Gold Plating	CU	50%	2
Unclear requirements or Changing requirements	PS	80%	2
Lack of user adoption or acceptance	DE	10%	2
Unable to manage risk	DE	30%	1
Security vulnerabilities or data breaches	PR	10%	1
End-users resist system	BU	40%	3
Delays is software developments or testing	CU	50%	3
Technical difficulties on system integration issues	BU	40%	2