



MOBILE APPLICATION

AGRICULTURE DEVELOPMENT

**By**

|  |  |
| --- | --- |
| Arwa bin kulaib | 434011491 |
| Shahad Aljathlani | 434004591 |
| Lama alashaq | 434006288 |
| Rawan ALmutairi | 434012030 |
| Manar abdelkarim | 434200161 |
| Fatima ALmotairi | 434009621 |

**Supervised By**

Dr.Lelia Mohamed Jamel

A Graduation Project Report Submitted to   
College of Computer Sciences and Information at PNU  
in Partial Fulfilment of the Requirements for the  
Degree of Bachelor of Science in Information System

CCIS, PNU

Riyadh, KSA

1439-1440H

# **DEDICATION**

This work is dedicated to our dear parents and friends.

# ACKNOWLEDGMENTS

We would like to express our deep appreciation, thanks and gratitude to our great teacher who helped us complete this report. Thanks to the special gratitude we give to our honorable Ms. Lelia Mohamed Jamel, who contributed to the mention of suggestions and encouragement, an effective discussion and a polite manner of supervising our project and introducing step by step.

Furthermore, we would also like to pay tribute to the gratitude extended to us for the success and excellence and the decisive role it has given us and to help us realize our dream.

# ABSTRACT

Farmers face difficulties in communicating with the Ministry of Agriculture, to get news and keeping track with events that are being held, also there is no direct link between customers and farmers, so they can’t sell their products directly to the customers.

This project provides a solution for to overcome these difficulties by creating mobile application that enables farmer and customer the register in the system and allows the farmer to direct sell his/her product to the customer.

Also, the application allows farmer to get notification in case there is an event will hold in the future and to receive all news that added by administrator.

The application in developed for android smartphones using android studio and java programming language.

# Keyword

Agriculture, farmers, mobile application, green grow

# TABLE OF CONTENTS

[1 DEDICATION i](#_Toc531158224)

[2 ACKNOWLEDGMENTS ii](#_Toc531158225)

[3 ABSTRACT iii](#_Toc531158226)

[4 Keyword iv](#_Toc531158227)

[5 TABLE OF CONTENTS v](#_Toc531158228)

[6 LIST OF FIGURES ix](#_Toc531158229)

[7 LIST OF TABLES xi](#_Toc531158230)

[1 CHAPTER 1: INTRODUCTION 2](#_Toc531158231)

[1.1 Introduction 2](#_Toc531158232)

[1.2 Proposal Solution 3](#_Toc531158233)

[1.3 The Project Goals 3](#_Toc531158234)

[1.4 The project aims 3](#_Toc531158235)

[1.5 Propose of this project 4](#_Toc531158236)

[1.6 Project Domain & Limitations 4](#_Toc531158237)

[2 CHAPTER 2: BACKGROUND INFORMATION 6](#_Toc531158238)

[2.1 Introduction 6](#_Toc531158239)

[2.2 Background Information 6](#_Toc531158240)

[2.3 The Techniques and Tools Used for Agriculture 6](#_Toc531158241)

[2.3.1 Development Mobile Application 6](#_Toc531158242)

[2.3.2 Programming language and design applied to agriculture development mobile application 7](#_Toc531158243)

[2.3.3 MySQL 7](#_Toc531158244)

[2.4 Methodology 7](#_Toc531158245)

[2.4.1 Waterfall Stages 7](#_Toc531158246)

[2.5 Waterfall Methodology 8](#_Toc531158247)

[2.5.1 Plantix 8](#_Toc531158248)

[2.5.2 Agri-Science Reference 9](#_Toc531158249)

[2.5.3 Ministry of Environment, Water and Agriculture 9](#_Toc531158250)

[2.6 Proposed and Similar Systems Comparison 10](#_Toc531158251)

[2.7 Conclusion 10](#_Toc531158252)

[3 CHAPTER 3: SYSTEM ANALYSIS 12](#_Toc531158253)

[3.1 Introduction 12](#_Toc531158254)

[3.2 System Analysis Overview 12](#_Toc531158255)

[3.3 Requirements Specification 12](#_Toc531158256)

[3.4 Description Of The Survey 13](#_Toc531158257)

[3.4.1 Questionnaire results 13](#_Toc531158258)

[3.5 Functional Requirements 13](#_Toc531158259)

[3.5.1 Employee of Ministry agriculture requirement 14](#_Toc531158260)

[3.5.2 Admin Requirements 15](#_Toc531158261)

[3.6 Non-functional requirements 15](#_Toc531158262)

[3.7 Requirement analysis 16](#_Toc531158263)

[3.7.1 Use case diagram 16](#_Toc531158264)

[3.7.2 Sequence diagram 21](#_Toc531158265)

[3.7.3 Class diagram 24](#_Toc531158266)

[4 CHAPTER 4: SYSTEM DESIGN 27](#_Toc531158267)

[4.1 Introduction 27](#_Toc531158268)

[4.2 System Architecture 27](#_Toc531158269)

[4.2.1 System modules 28](#_Toc531158270)

[4.3 User interface design 28](#_Toc531158271)

[4.4 Database Design 32](#_Toc531158272)

[4.5 Storyboard 34](#_Toc531158273)

[4.6 Conclusion 34](#_Toc531158274)

[5 CHAPTER 5: IMPLEMENTATION 36](#_Toc531158275)

[5.1 Introduction 36](#_Toc531158276)

[5.2 Implementation Steps 36](#_Toc531158277)

[5.3 Implementation Procedure 36](#_Toc531158278)

[5.3.1 Hardware requirements 36](#_Toc531158279)

[5.3.2 Software requirements 36](#_Toc531158280)

[5.4 Main Codes 37](#_Toc531158281)

[5.4.1 Login code 37](#_Toc531158282)

[5.4.2 Create Account Code 38](#_Toc531158283)

[5.4.3 Add Event Code 38](#_Toc531158284)

[5.4.4 View Cart Code 39](#_Toc531158285)

[5.5 Database Tables 42](#_Toc531158286)

[5.5.1 Admin table 42](#_Toc531158287)

[5.5.2 Farmer table 42](#_Toc531158288)

[5.5.3 Cutomer table 42](#_Toc531158289)

[5.5.4 Product table 42](#_Toc531158290)

[5.5.5 Category Table 43](#_Toc531158291)

[5.5.6 Event/News Table 43](#_Toc531158292)

[5.5.7 Cart Table 43](#_Toc531158293)

[5.5.8 Oerder Table 43](#_Toc531158294)

[5.6 Screenshots 43](#_Toc531158295)

[6 CHAPTER 6: TESTING 61](#_Toc531158296)

[6.1 Introduction 61](#_Toc531158297)

[6.2 Testing types 61](#_Toc531158298)

[6.2.1 White box testing 61](#_Toc531158299)

[6.2.2 Black box testing 61](#_Toc531158300)

[6.2.3 Functional requirements 61](#_Toc531158301)

[6.2.4 Unit testing 61](#_Toc531158302)

[6.2.5 System testing 61](#_Toc531158303)

[6.3 Testing Plan 62](#_Toc531158304)

[6.3.1 Admin Functionalities 62](#_Toc531158305)

[6.3.2 Farmer Functionalities 62](#_Toc531158306)

[6.3.3 Customer Functionalities 63](#_Toc531158307)

[6.4 Test Cases 63](#_Toc531158308)

[6.4.1 Admin Side 63](#_Toc531158309)

[6.4.2 Farmer Side 64](#_Toc531158310)

[6.4.3 Customer side 65](#_Toc531158311)

[6.5 Testing results 66](#_Toc531158312)

[6.6 Results 67](#_Toc531158313)

[7 CHAPTER 7: CONCLUSION AND FUTURE WORK 69](#_Toc531158314)

[7.1 Conclusion 69](#_Toc531158315)

[7.2 Future works 69](#_Toc531158316)

[8 Appendix (A) 70](#_Toc531158317)

[8.1 Questionnaire for agriculture 70](#_Toc531158318)

[9 References 75](#_Toc531158319)

# LIST OF FIGURES

[Figure ‎2.1 Waterfall methodology 8](#_Toc531158070)

[Figure ‎2.2 Plantix App 9](#_Toc531158071)

[Figure ‎2.3 Agri-Science Reference 9](#_Toc531158072)

[Figure ‎2.4 Ministry of Environment, Water and Agriculture 9](#_Toc531158073)

[Figure ‎3.1 Use case diagram 16](#_Toc531158074)

[Figure ‎3.2 Register Subsystem Use Case diagram 17](#_Toc531158075)

[Figure ‎3.3 Buying subsystem Use case diagram 18](#_Toc531158076)

[Figure ‎3.4 Manage Item subsystem use case diagram 19](#_Toc531158077)

[Figure ‎3.5 Add Information subsystem Use case diagram 20](#_Toc531158078)

[Figure ‎3.6 chat subsystem use case diagram 21](#_Toc531158079)

[Figure ‎3.7 Sequence Diagram of Use Case "Register" 22](#_Toc531158080)

[Figure ‎3.8 Sequence Diagram of The Use Case "Buy" 23](#_Toc531158081)

[Figure ‎3.9 Sequence Diagram of the use case "Add Information" 23](#_Toc531158082)

[Figure ‎3.10 Sequence Diagram of Use Case “Chat 24](#_Toc531158083)

[Figure ‎3.11 Grow Green Application Class Diagram 25](#_Toc531158084)

[Figure ‎4.1 System architecture 27](#_Toc531158085)

[Figure ‎4.2 Login interface 28](#_Toc531158086)

[Figure ‎4.3 : the registration page 29](#_Toc531158087)

[Figure ‎4.4 Home page 29](#_Toc531158088)

[Figure ‎4.5 user profile 30](#_Toc531158089)

[Figure ‎4.6 edit profile 30](#_Toc531158090)

[Figure ‎4.7 chat screen 31](#_Toc531158091)

[Figure ‎4.8 products page interface 31](#_Toc531158092)

[Figure ‎4.9 search screen 32](#_Toc531158093)

[Figure ‎4.10 database farmar 32](#_Toc531158094)

[Figure ‎4.11 database product 33](#_Toc531158095)

[Figure ‎4.12 database customer 33](#_Toc531158096)

[Figure ‎4.13 application storyboard 34](#_Toc531158097)

[Figure ‎5.1 Splash screen 44](#_Toc531158098)

[Figure ‎5.2 Main screen 44](#_Toc531158099)

[Figure ‎5.3 About screen 45](#_Toc531158100)

[Figure ‎5.4 Registration screen 45](#_Toc531158101)

[Figure ‎5.5 Login screen 46](#_Toc531158102)

[Figure ‎5.6 Reset password screen 46](#_Toc531158103)

[Figure ‎5.7 Admin main screen 47](#_Toc531158104)

[Figure ‎5.8 events screen 47](#_Toc531158105)

[Figure ‎5.9 add event screen 48](#_Toc531158106)

[Figure ‎5.10 news screen 48](#_Toc531158107)

[Figure ‎5.11 add news screen 49](#_Toc531158108)

[Figure ‎5.12 Change password screen 49](#_Toc531158109)

[Figure ‎5.13 Farmer main screen 50](#_Toc531158110)

[Figure ‎5.14 Products list screen 50](#_Toc531158111)

[Figure ‎5.15 add new product 51](#_Toc531158112)

[Figure ‎5.16 Events screen 51](#_Toc531158113)

[Figure ‎5.17 Event discussions 52](#_Toc531158114)

[Figure ‎5.18 Change password screen 52](#_Toc531158115)

[Figure ‎5.19 Edit profile 53](#_Toc531158116)

[Figure ‎5.20 Order screen 53](#_Toc531158117)

[Figure ‎5.21 Customer main screen 54](#_Toc531158118)

[Figure ‎5.22 categories screen 54](#_Toc531158119)

[Figure ‎5.23 Product evaluations 55](#_Toc531158120)

[Figure ‎5.24 add product to cart screen 55](#_Toc531158121)

[Figure ‎5.25 Add evaluation screen 56](#_Toc531158122)

[Figure ‎5.26 cart screen 56](#_Toc531158123)

[Figure ‎5.27 Payment screen 57](#_Toc531158124)

[Figure ‎5.28 customer orders list 57](#_Toc531158125)

[Figure ‎5.29 Change password screen 58](#_Toc531158126)

[Figure ‎5.30 Edit profile screen 58](#_Toc531158127)

# LIST OF TABLES

[Table ‎2.1 Proposed and Similar Systems Comparison 10](#_Toc530230105)

[Table ‎3.1 Agriculture Mobile Application use case diagram description 17](#_Toc530230106)

[Table ‎3.2 Register Subsystem use case diagram description 18](#_Toc530230107)

[Table ‎3.3 Buying subsystem use case diagram description 18](#_Toc530230108)

[Table ‎3.4 Masnage item subsystem use case description 19](#_Toc530230109)

[Table ‎3.5 Add Information subsystem Use case des 20](#_Toc530230110)

[Table ‎3.6 chat subsystem use case diagram description 21](#_Toc530230111)

[Table ‎6.1 Admin side test cases 63](#_Toc530230112)

[Table ‎6.2 farmer side test cases 64](#_Toc530230113)

[Table ‎6.3 Customer test cases 65](#_Toc530230114)

[Table ‎6.4 Testing results 66](#_Toc530230115)

Chapter 1

INTRODUCTION

# CHAPTER 1: INTRODUCTION

## Introduction

Agriculture is one of the main industries in Saudi Arabia. It focuses on exporting dates, dairy products, eggs, fish, poultry, fruits, vegetables, and flowers to markets around the world. The government of Saudi Arabia is heavily involved in the agriculture industry. The government offers long-term interest-free loans and low-cost water, fuel, electricity, and duty-free imports of raw materials and machinery. The ministry of environment, water and agriculture is primarily responsible for the agricultural policies in Saudi Arabia.

Technology is widely connected to all industries including agriculture. Advancement of information technology facilitates time-consuming processes by transforming them into an easy and quick process. One of the application areas of information and communication technologies that facilitates time-consuming processes is e-agriculture. E-agriculture ensures the systematic dissemination of information using ICTs on agriculture, animal husbandry, fisheries, forestry, and food in order to provide an access to inclusive, up-to-date, and detailed knowledge and information, particularly in rural areas. Public-private partnerships should increase the use of ICTs as tools that improve production (quantity and quality).

Since that the agriculture is an important area in developing the national economic, the project aims to participate in improving and developing this field. Spreading information and sharing knowledge is an important step in the process of improving agriculture. The application helps farmers to develop, work, sell products without relying or depending on anyone but themselves. In order to do that, this project provided an application that includes a list of information about agriculture with detailed descriptions of each agricultural courses, farmlands, products, etc. also. Problem Statement & Significance

1. The agriculture in Saudi Arabia has many problems and issues, some of them are global issues. The main issues are:
2. The desert environment, the affecting of seasonal diseases on plants and animals, the lack of good quality water, and soil salinity.
3. The use of chemical and insecticides on plants and not caring of customers' health.
4. The corner of the market by the foreign residents and imported goods, coupled with weakness at the marketing of the local products outside of Saudi Arabia and customer’s preference of the imported goods.
5. Farmers have no incentive or motivation for development and improvement so that they are still using the old, traditional methods
6. The small income for farmers (as a result of the need for a mediator) especially compared to the price of their goods in the market.
7. The lack of exchange of experiences between Saudi farmers.
8. This application helps the farmers to solve these problems.

## Proposal Solution

In the light of the previous problems the extent of the problems of agriculture and its effects on the plant have been realized, and some solutions have been set which benefit the farmers, including:

1. Ministry of environment, water and agriculture.
2. Agricultural development fund.
3. Agricultural extension of seasonal diseases "plan tix"
4. Training courses

## The Project Goals

The actual project aims to rich following goals

1. Provide online shopping.
2. Provide a rural community development and the development of services and facilities.
3. Help farmers produce better types of crops to consumers and their families.
4. Improve the standard of living of farmers and improve their economy by increasing agricultural production and thus increase their income.
5. Introduce the different agricultural production supplies commensurate with their potential and abilities.
6. Involve farmers and invite them at events, seminars and meetings of various agricultural their benefit by acquiring new skills through discussion and the meeting and seen.

## The project aims

This project some specific aims are to be reached:

1. Learning how to design an application that helps and saves farmers time.
2. Learning how to analyze the problem and identify appropriate solutions.
3. Learning programming languages ​​and how to use them such as: "php", "java", "c ++", "html".

## Propose of this project

The application is like a link between the farmer and the buyer under the supervision of the ministry of agriculture. This application is a reference record for the farmer and the buyer through which the agricultural products are improved and the sales process is improved to customers

1. Provides farmers with information of seasonal diseases affecting plants and vaccines.
2. Provides farmers with information of weather, climate and agriculture seasons.
3. Improves agricultural products through the provision of fertilizer and arable soil
4. Provides assistance and advice by farmers' specialists in improving their products
5. Helps farmers with a good crop provide employment to help them in agriculture
6. Provides educational courses and free consultations through application to farmers.
7. Helps farmers to product by selling their products directly to customers and markets.

## Project Domain & Limitations

This project provides mobile application for farms management; especially for farms in Riyadh, Saudi Arabia. This application is used by customers and farms.

Chapter 2

BACKGROUND INFORMATION

# CHAPTER 2: BACKGROUND INFORMATION

## Introduction

In order to build a new system, it is necessary to first study and understand what has been discussed before about building systems, the current problems, the method of solving these problems, the proposed solutions, and technologies or techniques required to find a solution. In this paper, these points are to be covered in some details. Section 2.2 provides background information; this is followed by Section 2.3, which explains the related survey. Then, Section 2.4 shows a table of proposed systems and comparison of similar systems.

## Background Information

The following sections discuss some information about the techniques and tools used in building this application, the programming languages, and the designing programs that are applied.

## The Techniques and Tools Used for Agriculture

### Development Mobile Application

UML (Unified Modeling Language) is a language for visualizing, specifying, constructing, and documenting the software system. [1] UML is made up of notation and diagrams. [1] Notation consists of the elements that work together in a diagram, such as symbols, connectors, notes, values, codes, or pseudo codes. Diagrams are pictorial representations of a process, a system or some part of it. UML is defined as a set of specifications created and distributed by the Object

Management Group (OMG). [2] UML diagram can be divided into two categories:

1. Structure diagrams are used in documenting the architecture of software system and are involved in the system being modelled such as class diagram.
2. Behavior diagrams represent functionality of software system and emphasize what must happen in the system being modelled. There are different behavior diagrams, which are:
   1. Use Case Diagrams (UCDs): They are widely used to describe requirements and desired functionality of software products.
   2. Sequence Diagrams: They include more than just events, their focus on the time-ordered sequence of transactions. [3]

### Programming language and design applied to agriculture development mobile application

#### PHP

The first programming language is PHP, which is an acronym for Personal Home Page. It is called so because lots of programmers used it to build their websites, going much further than what was possible with HTML, CSS, and JavaScript. PHP is a server-side, HTML-embedded scripting language that may be used to create dynamic Web pages. It is available for most operating systems and Web servers; it also can access most common databases, including MySQL. PHP may be run as a separate program or compiled as a module for use with a Web server. It is used in this project to link the database with the mobile application. [4]

#### Phone Gap/Cordova

It is a framework that enables the enterprise to target multiple smartphone platforms with a single code base using technologies the enterprise is already largely familiar with. It is a perfect fit for the enterprise, as it does not require duplication of effort to build multiple native applications that must essentially be rewritten for each supported platform. PhoneGap application can do anything that can be coded in standard, everyday HTML, CSS, and JavaScript. PhoneGap is the framework tool for this application. [5]

### **MySQL**

In this application, there is a need to database to store data. Therefore, MySQL was the appropriate choice as a database for this project MySQL stores the user’s information, the customers information, and anything else the user wants to store.[4]

## Methodology

The methodology that is used in this project was a waterfall as shown in figure 2.1

The waterfall model is a classical model used in system development life cycle to create a system with a linear and sequential approach.[10]

### Waterfall Stages

1. Requirements: All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
2. **Design**: The requirement specifications from first phase are studied in this phase; and the system design is prepared. The system design helps in specifying hardware and system requirements; as it also helps in defining the overall system architecture.
3. **Implementation:**When the design is fully completed, an implementation of that design is made by coders. Towards the later stages of this implementation phase, system components produced by different teams are integrated.
4. **Verification:** After the implementation and integration phases are completed, the system is tested and debugged; any faults introduced in earlier phases are removed here.
5. **Maintenance:** The function of the system should be maintained to ensure if any error or bugs occurred, that they can be fixed.

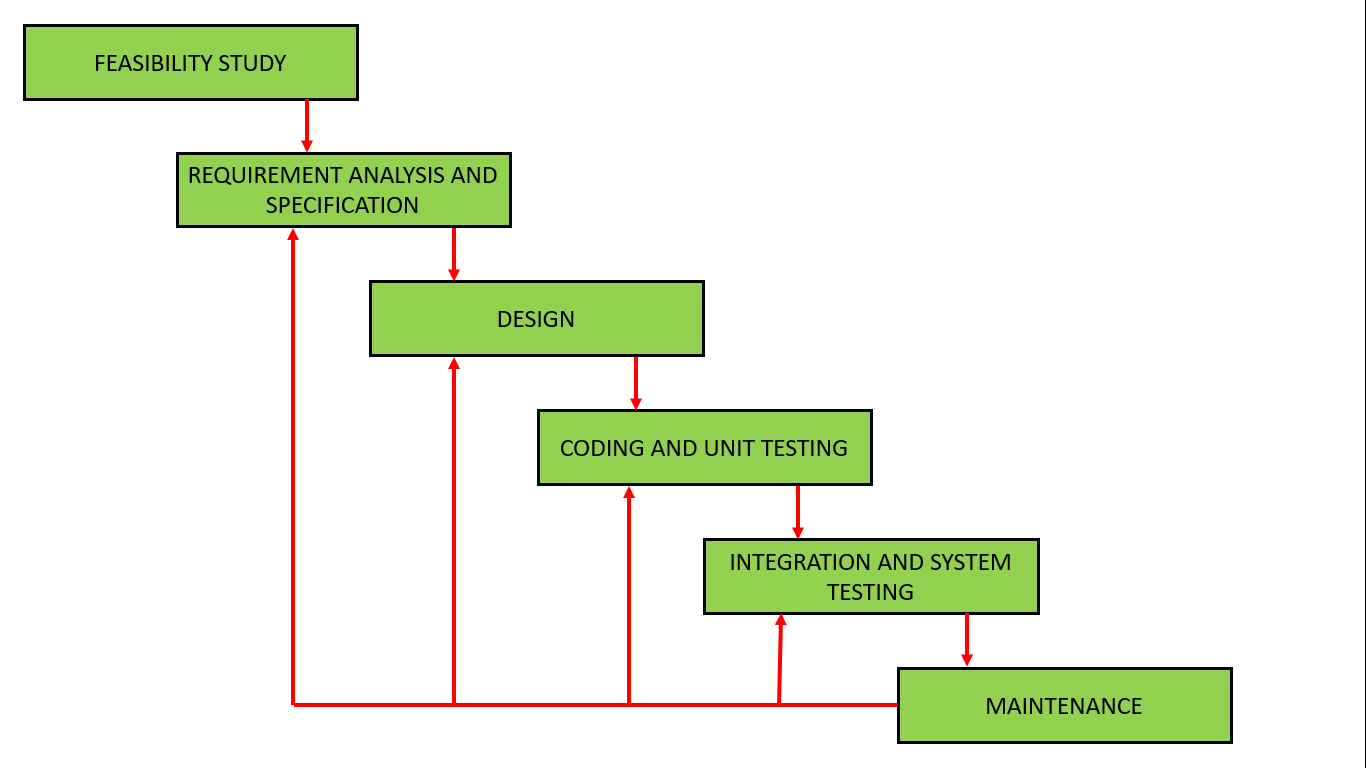


Figure ‎2.1 Waterfall methodology

## Waterfall Methodology

The systems being discussed are related to GreenGrow that was used in building agriculture development mobile application. The main elements of this application include plan tix, Agri-Science Reference, and Ministry of Environment, Water and Agriculture. The following is a brief description of each

### Plantix

It is an application that enables farmers, workers, and gardeners around the world to diagnose plant diseases, pests, and nutrient deficiencies affecting their crops. All they need to do is just to take a simple smartphone photo and upload it to the application. image recognition is able to detect more than 240 plant pests’ diseases automatically. Then, the application offers to keep them in touch with a large community for exchanging knowledge about topics like crop cultivation, disease control, and best practice.[6]



Figure ‎2.2 Plantix App

### Agri-Science Reference

It is a website to share reference files in all branches of agricultural sciences (such as agro-chemistry, agronomy, fertilizers, and fertilization, etc.). In order to facilitate the access of agriculture's interested beginners and experts to references needed to complete their applied work or scientific research[7].



Figure ‎2.3 Agri-Science Reference

### Ministry of Environment, Water and Agriculture

The ministry of Environment, water and Agriculture provided an application that has services enable users to review the latest bill. it also enables citizens to communicate their voice to the officials of the ministry of environment, water and agriculture, and inquire about the application as well as their transactions in the ministry[8].



Figure ‎2.4 Ministry of Environment, Water and Agriculture

## Proposed and Similar Systems Comparison

Table ‎2.1 Proposed and Similar Systems Comparison

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Applications and websites | GreenGrow | Plantix | Ministry of Environment, Water and Agriculture our project | Agri-Science Reference |
| language | Arabic | English | Arabic | Arabic |
| Operating System | ios | Android | Android  & ios | Android |
| Audience | People involved in agriculture in Saudi Arabia such as farmers and costumers | Germany and Switzerland agriculture | People involved in agriculture in Saudi Arabia such as farmers and costumers | Arab countries agriculture |
| Training | Yes | \_ | Yes | \_ |
| Sale | Yes | \_ | \_ | \_ |
| Chatting | Yes | \_ | Yes | \_ |
| Payment | Cash - online payment | \_ | \_ | \_ |

## Conclusion

This paper began with the background. It listed some of the related work survey for this application. Finally, this paper discussed the proposed and similar system comparison.

Chapter 3

SYSTEM ANALYSIS

# CHAPTER 3: SYSTEM ANALYSIS

## Introduction

Analysis is the process of extracting. Compiling. And modeling raw data for purposes of obtaining information that can be applied to formulating conclusions, predicting outcomes or supporting decisions in business, scientific and social science settings. This chapter explains how users' requirements were collected what their requirements and other requirements which are functional, nonfunctional, Hardware and software requirements.

## System Analysis Overview

System Analysis is the process of studying a procedure or business in order to identify its goals and purposes and create systems and procedures that will achieve them in an efficient way.[9]

A problem -solving technique that decomposes a system into its component pieces for the purpose of studying how well those component parts work interacts to accomplish their purpose

an inquiry to aid the decision maker to choose a course of action by systematically investigating his proper objectives, comparing quantitatively wherever possible costs effectiveness and risks associated with the alternative policies or strategies for achieving them

The purpose of analysis process is to transform the needs and high-level requirements specified in earlier into unambiguous (measurable and testable) traceable, complete, consistent, and stakeholder approved requirements which help later in building a logical model of the new system

The system analysis importance is to help the analyst to identify the problem, know the parts and the functions of different components and provide effective communication between user and analyst. The analyst must involve the user in all stages of the systems analysis process.

## Requirements Specification

A questionnaire has been chosen to gather the opinions of customers and farmers to define the problems and gather requirements for an optimal agriculture mobile app. The survey contains multiple-choice questions.

## Description Of The Survey

This questionnaire was used as way to collect the opinions of farmers, and customers about the access to agriculture events , requested services from the ministry of agriculture and requested training by using google questionnaire.

The details of the survey are given in appendix A.

### Questionnaire results

The survey has been sent to 30 people, but has answered 25 people, 80% are farmers and 20% are customers .

Q1: What is the best purchase of the product from the application or from the shop where the farmer sells it?

Result: 72% farmer shop

28% application

Q2: What would you prefer to have in the app?

Result: 40% to be linked to the Ministry of Agriculture.

20% Contains information on agriculture.

40% Contains updated information on new seasonal diseases and methods of prevention and treatment.

Q3) What is better to use an application linked to the Ministry of Agriculture or go to the headquarters of agriculture?

Result: 72% application

28% headquarter of the ministry.

Q4) Do you prefer to enroll in training courses?

Result: 72% yes

28% no

Q5) Which courses do you prefer to enroll in?

Result: 64% Learning on new farming techniques

36% Pest Control Technology for Agricultural Crops

Q6) Would you prefer to share your information with other farmers through the app?

Result: 72% yes

28% no

## Functional Requirements

Functional requirements define a function of a system or its component.

1. Register for a New Account and Login

The system allows farmers to register by filling a registration form and submitting it.

The system will verify the information, create account and register in users table. Then the user can be logged in

1. View and Edit profile

The system allows the user to view its information and be ability to modify and save all changes**.**

1. Follow and Unfollow

The farmer can also follow-up and cancel the follow-up to the news page and can put an alert to all the news that interests him.

1. View Even

The system allows the farmer to view all the events that are provided by the Ministry of agriculture to farmers and view all the News on agricultural crops and weather conditions , manures , diseases.

1. Chatting
   1. The site provides communication chat service between the farmers and between customers / farmers .
   2. To sell products online.
   3. To have payment online.

### Employee of Ministry agriculture requirement

1. Register for new account and login

The application allows employee of ministry agriculture to register by filling a registration form and submitting it. The application verifies the information, create account and register them in ministry of agriculture employee table. Then can be login.

1. View and edit profile

The application allows the employee of ministry agriculture to view its information and be able to modify them and save all change.

1. View and Manage Event

The application enables the employee of ministry agriculture to view all events that he had published, and also enable the employee to manage the events (edit, delete and post)

1. Logout

### Admin Requirements

1. Admin is able to login
2. Admin is able to logout
3. Admin is able to manage Farmers by adding, editing, deleting
4. Admin is able to manage sales schedule by adding, editing, deleting.
5. Admin is able to manage comments by adding, editing, deleting.
6. Admin is able to change person's password
7. Admin is able to update all users' privileges
8. Admin is able to manage gallery images by adding editing. Deleting
9. Admin is able to manage news and events by adding, editing deleting

## Non-functional requirements

Non-functional requirements are requirements that specify criteria that can be used to describe how the system works and what the properties of the system is.

GreenGrow app has some important non-functional requirement as the following:

1. Reliability : users must be trusting the system; the system must be work fine for long time and does not affected by the number of users.
2. Performance: the system must response to the users request very fast.
3. Availability: the system must be online all the time. It called uptime, and nowadays, the uptime of the online systems reaches 99.8%.
4. Security: as this system has personal and private data of the users, it must provide high security level to prevent any one who didn’t have permissions to change the system data or get users information.
5. Usability: the system shall be easy to use, and support Arabic languages.
6. Extensibility: the system must be ready for adding new features.
7. Maintainability: the system should be easy to maintain and fixed.
8. Reliability: the system must provide reliable information for system users.
9. Documented: the system must be full documented specially the code.

## Requirement analysis

System requirements are descriptions of the services that a software system must provide, and the constraints under which it must operate [1].

Any system has two types of requirements, functional requirements and non-functional requirements. This section describes the results of analyzing the requirements to extract information as a

structured list of requirements. So, based on the object-oriented approach, will should include the following Diagram:

1. Use Case Diagram.
2. Sequence Diagram.
3. Class Diagram.

All these diagrams have been drawn using diagrams online tool "drow Io diagrams" program.

### Use case diagram

Use case diagrams describe what a system does to benefit users. A use case model is built through an iterative process, lead to a requirement specification on which all agree. [1]

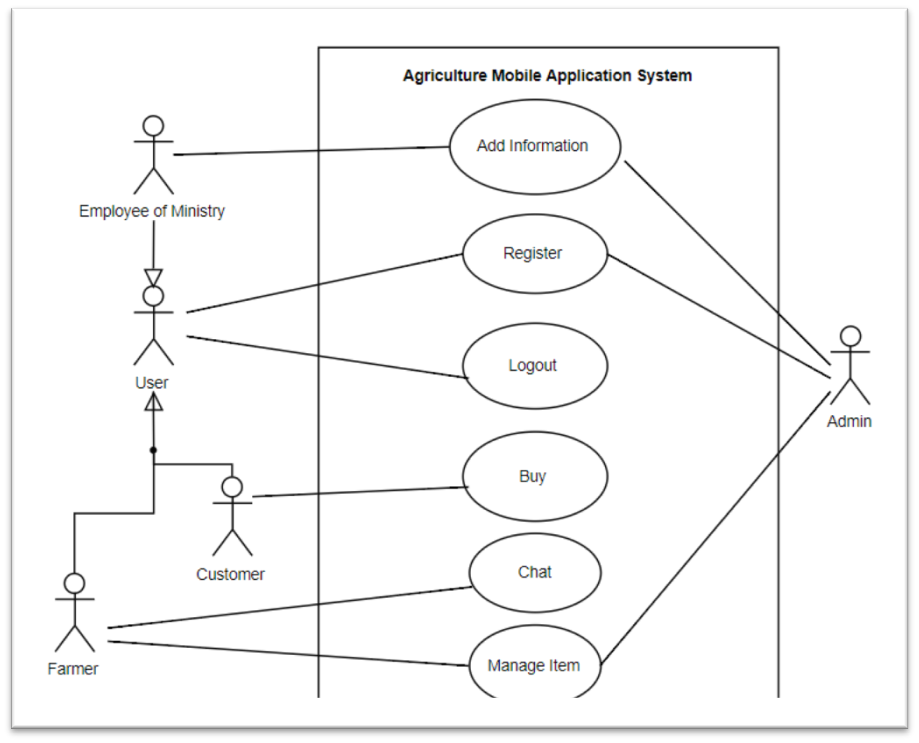


Figure ‎3.1 Use case diagram

Table ‎3.1 Agriculture Mobile Application use case diagram description

|  |  |
| --- | --- |
| Use Case: | Agriculture Mobile Application |
| Primary actor: | Admin , customer, farmer, employee of ministry |
| Descriptions: | This use case captured Green Grow system. The user can register as an Admin or a customer or a farmer, or employee of ministry then he/she can buy or sell or view news and events or manage news and events or chatting according to his privilege and which type of user is he. |
| Pre- conditions: | Must have an active internet connection. |

#### Register Use case

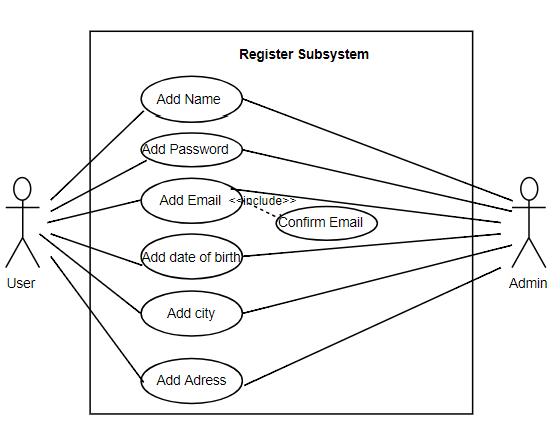


Figure ‎3.2 Register Subsystem Use Case diagram

Table ‎3.2 Register Subsystem use case diagram description

|  |  |
| --- | --- |
| Use Case: | Register subsystem |
| Primary actor: | Admin , user |
| Descriptions: | User creates new account to apply an order, user enter the application  - User need to provide his/her information [username, password,  email, phone number]  - The system check if all information is valid then complete |
| Pre- conditions: | -The user must access the application  - The user should complete all the requested field  - The user should provide correct information. |
| Post-conditions: | The registration finishes successfully. |

#### Buy Use Case

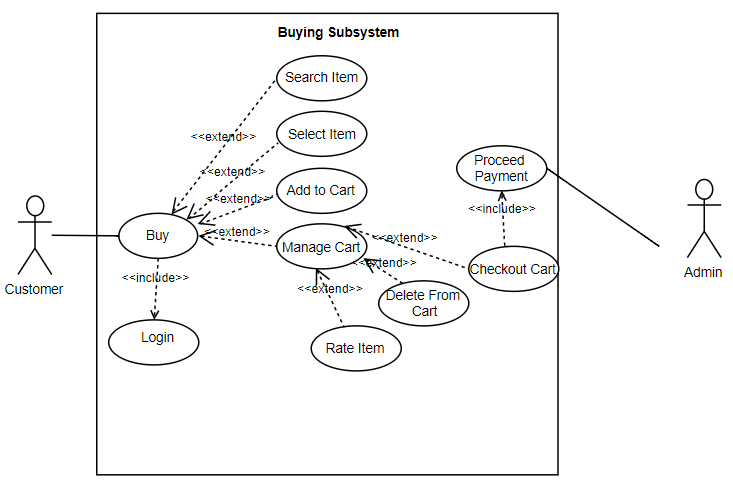


Figure ‎3.3 Buying subsystem Use case diagram

Table ‎3.3 Buying subsystem use case diagram description

|  |  |
| --- | --- |
| Use Case: | Buying system |
| Primary actor: | Admin , customer |
| Descriptions: | This use case describes how a user can Buy in the application |
| Pre- conditions: | Must login  Must have an active internet connection. |
| Post-conditions: | The customer has processed an order for their desired items successfully. |

#### Manage item subsystem Use case

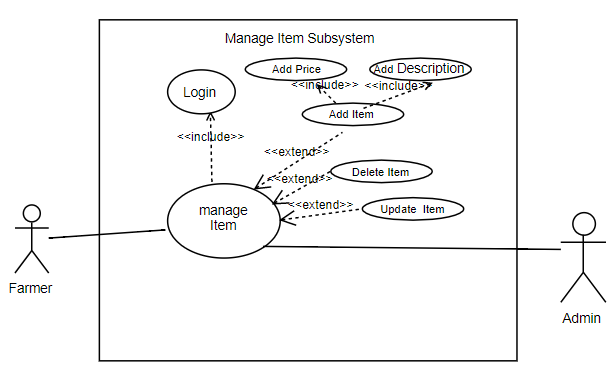


Figure ‎3.4 Manage Item subsystem use case diagram

Table ‎3.4 Masnage item subsystem use case description

|  |  |
| --- | --- |
| Use Case: | Manage Item Subsystem |
| Primary actor: | Admin , farmer |
| Descriptions: | The system displays managing Item process, the farmer can add, delete and update items. The admin has privilege to manage items. |
| Pre- conditions: | Must login be using a farmer account  Must have an active internet connection. |
| Post-conditions: | The registration finishes successfully |

#### Add Information Use Case

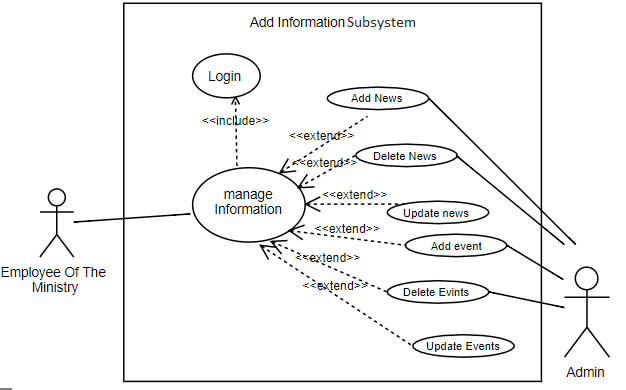


Figure ‎3.5 Add Information subsystem Use case diagram

Table ‎3.5 Add Information subsystem Use case des

|  |  |
| --- | --- |
| Use Case: | Add information subsystem. |
| Primary actor: | Admin, employee of the ministry. |
| Descriptions: | The employee of ministry manages event. Admin can add and delete events and news. |
| Pre- conditions: | Must have an active internet connection. |
| Post-conditions: | The farmer received news and events successfully. |

#### Chat Use Case

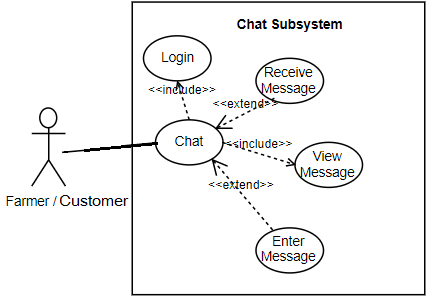


Figure ‎3.6 chat subsystem use case diagram

Table ‎3.6 chat subsystem use case diagram description

|  |  |
| --- | --- |
| Use Case: | Chatting system |
| Primary actor: | Farmer , customer |
| Descriptions: | Describe the process of chatting |
| Pre- conditions: | Must have an active internet connection.  Must login with farmer or customer account. |
| Post-conditions: | Farmers/customers can chat with each other |

### Sequence diagram

Sequence diagram shows objects interacting along lifelines that represent general order. [1] The use cases described in section 3.2.2 are more detailed in the section by using sequence diagram.

#### Sequence Diagram of Use Case "Register"

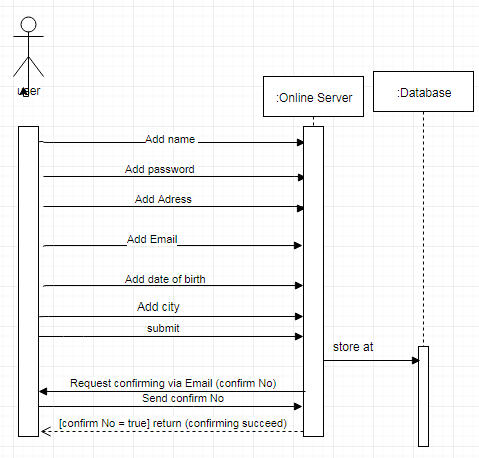


Figure ‎3.7 Sequence Diagram of Use Case "Register"

#### Sequence Diagram of The Use Case "Buy"

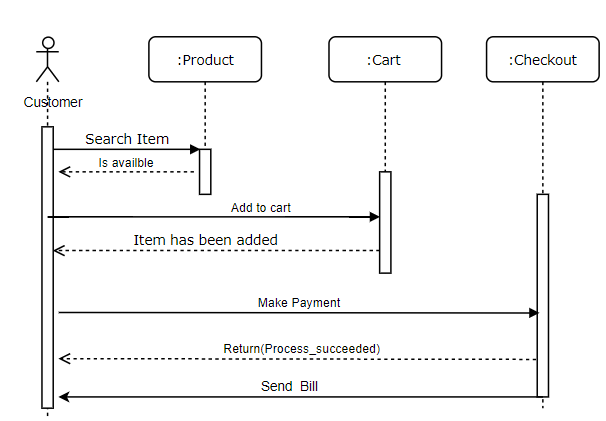


Figure ‎3.8 Sequence Diagram of The Use Case "Buy"

#### Sequence Diagram of the Use Case "Add Information"

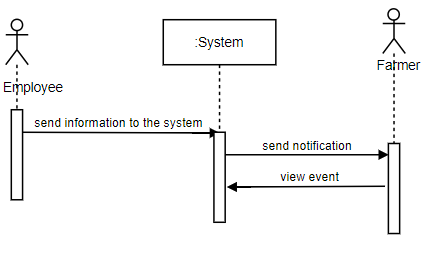


Figure ‎3.9 Sequence Diagram of the use case "Add Information"

#### Sequence Diagram of Use Case “Chat”

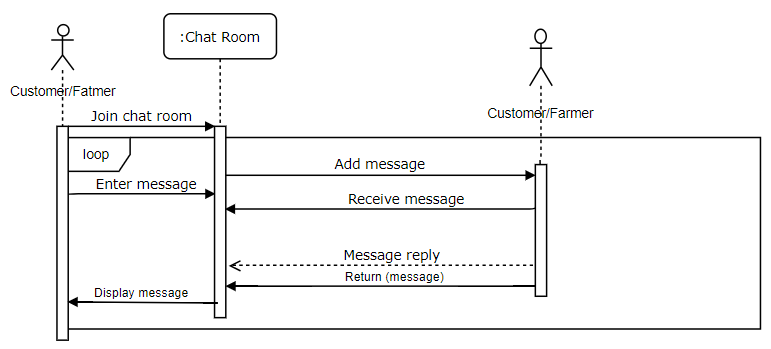


Figure ‎3.10 Sequence Diagram of Use Case “Chat

### Class diagram

A class diagram describes the static view (structure) of a system. It shows the entities involved in the system in their relationship [1].

This section shows all the classes created for the application and the relationships between these classes as shown in figure 3.11

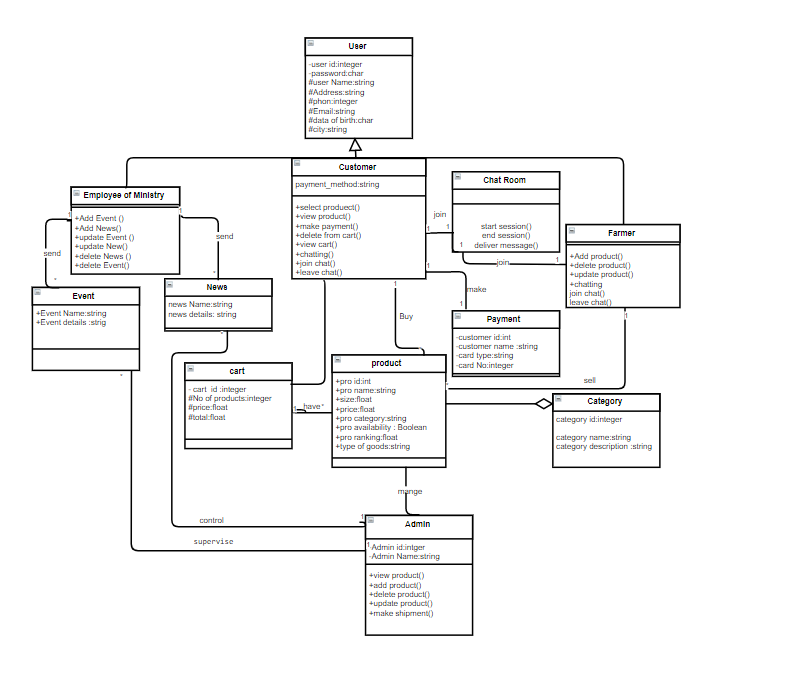


Figure ‎3.11 Grow Green Application Class Diagram

Chapter 4

SYSTEM DESIGN

# CHAPTER 4: SYSTEM DESIGN

## Introduction

Systems design simplifies the design of systems. It implies a systematic and rigorous approach to design – an approach demand by the scale and complexity of many systems problems. Today, ideas from design methods and systems design may be more relevant to designers than ever before – as more and more designers collaborate on designing software and complex information spaces. Frameworks suggested by systems design are especially use full in modeling interaction and conversation. They are also useful in modeling the design process itself.

## System Architecture

Our system is three-tier architecture that consists of three machines:

**Presentation tier:** This is the top level of the application. Include the presence of the internet,also a mobile app interface to facilitate user access to the system features and services. (It is a layer which users can access directly)

**Middle tier:** Also, called the application tier, logic tier, business logic or logic tier, this tier is pulled from the presentation tier. It controls application functionality by performing detailed processing.

**Data storage tier:** This layer consists of database server managed by the admin and stores all system data.

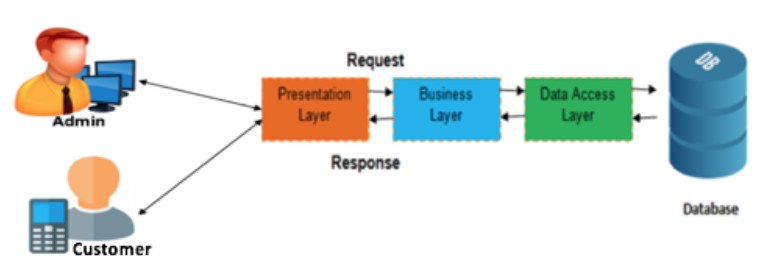


Figure ‎4.1 System architecture

### System modules

This system consists of two main modules the admin side and the customer side:

**Admin side:** It’s the module where the admin can manage the system; it is an internet-based application. The administrator can enter this module by his/her username and password then he can add user, delete user, manage user’s accounts and activities.

**Customer side:** It's a mobile application where the user can make many activities according to his/her privilege such as buying , selling , adding news and events , view news and events and mange his/her account.

## User interface design

This section proposes the most important interfaces of Green Grow application.

If the customer wants to login to the system, he/she needs to enter his/her name and password like what’s appeared in the figure 4-2 "login interface

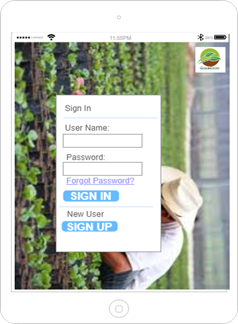


Figure ‎4.2 Login interface

The figure 4-2 sign in interface where the user enters username, password to access.

If the user has no account "if he/she is a new user", he/she can click on sign up .



Figure ‎4.3 : the registration page

The figure 4-3 shows the registration page , where the user must enter her or his information to complete the registration process and, he should choose if it is a farmer , customer, or ministry employee.

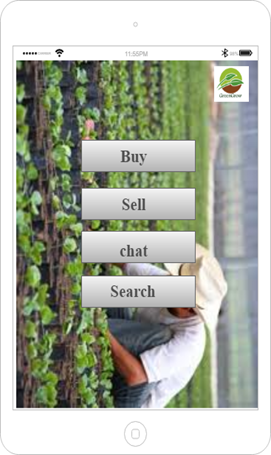


Figure ‎4.4 Home page

Figure 4.4 home page for Green Grow displays the buy, sell, chat and search

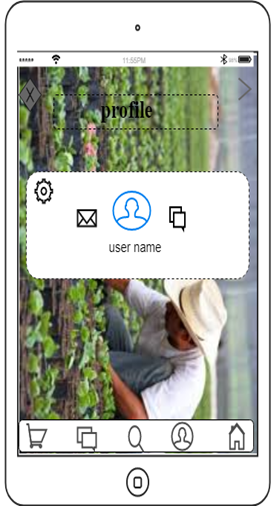


Figure ‎4.5 user profile

The figure 4-5 shows the user profile, which displays username, profile picture , edit profile icon , chat , and shopping cart.

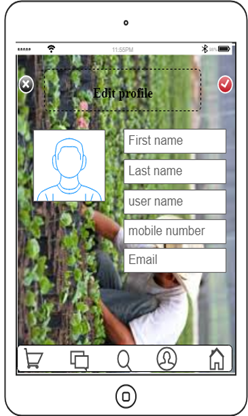


Figure ‎4.6 edit profile

The figure 4-6 shows the displayed interface page if the user clicked the edit profile icon. The user can edit his/her name, mobile number, and email.



Figure ‎4.7 chat screen

The figure 4-7 chat screen: it contains the messages sent and received between the customers /farmers.



Figure ‎4.8 products page interface

Figure 4.8 shows the product interface that contains the product image, name, price, quantity, and the option to add the product directly to the shopping cart.



Figure ‎4.9 search screen

Figure 4.9 search screen: it contains a search box to search for farmer by his/her username, as well as a list of the products categories to search for.

## Database Design

The database is an organization of the data set. It is a set of charts and tables, queries, reports, views, and other objects. A database we can perform an update, manage and link the data is accurate, secure and easy. A database in our system consist of four tables (user, customer, and farmer)

The application manages and store data in database using MYSQL as shown below:



Figure ‎4.10 database farmar

The table contains information about the farmer .

It contains the following columns: No: the row contains the table primary key and works with (auto increment) that each record is inserted to the table and has a serial number,

1. Name: contains the brief name of the student that is viewed.
2. Username: contains the user id which used to login in to system.
3. Password: contains the password of the student.
4. Email: contains the email to the student.
5. Birthdate: contains the date birth of student .
6. 5-Mobile: contains the mobile of farmer.
7. Age: contains the age of the farmer

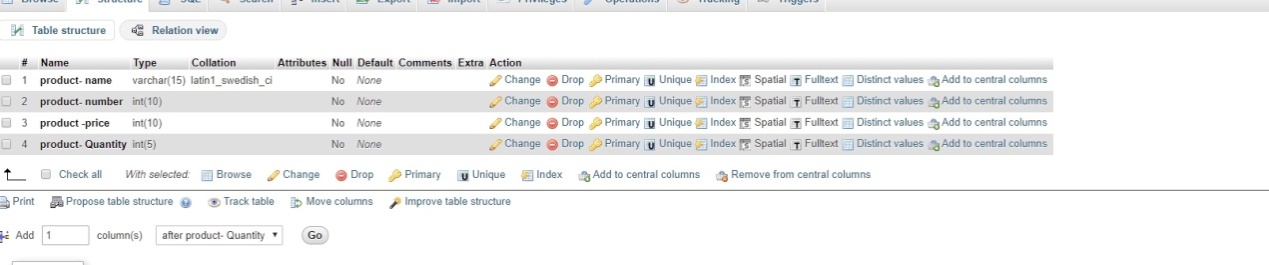


Figure ‎4.11 database product

The table contains information about the product.

1. Product\_number: the row contains the primary key of the table works with (auto increment) that each record inserted to the table take a serial number to be unique in its value.
2. Product\_name: contains the name of products.
3. Product\_price: contains the prices of products.
4. Product\_quantity: contains the quantity of the products.



Figure ‎4.12 database customer

The table contains information about the customer.

1. Customer\_number: the row contains the primary key of the table works with (auto increment) that each record inserted to the table take a serial number to be unique in its value.
2. First\_name: contains customer first name.
3. Last\_name: contains customer last name.
4. Mobile\_number: the customer own mobile phone number.
5. Username: contains customer log in name that is unique.
6. Password: the customer own secret series of numbers or letters.
7. Email: the electronic mail of the customer.

## Storyboard

A storyboard is a visual representation of the application structure. lt out all the components of the application and how they inter-relate. creating a application storyboard can help to plan and organize the application and even plan the internal linking structure between pages.the figure (4.13) show application storyboard

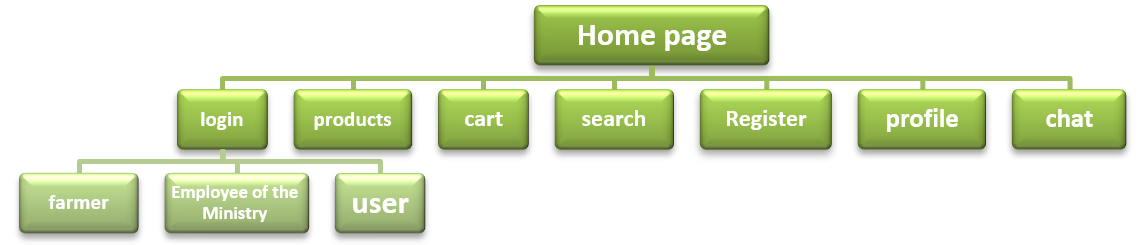


Figure ‎4.13 application storyboard

## Conclusion

This chapter clarified the project idea, such as the presentation and the pictures "interfaces" that show the form of the application, discussed the elements of the system design, defined the system architecture associated with GreenGrow application by defining the layers and the functions, described the application's data in the database.

Chapter 5

SYSTEM IMPLEMENTATION

# CHAPTER 5: IMPLEMENTATION

## Introduction

System implementation is the 4th phase in the system life cycle for successful new system design. The project takes shape during the implementation phase. This phase involves the construction of the actual project result. It is during this phase that the project becomes visible to outsiders, to whom it may appear that the project has just begun. The implementation phase is the doing phase, and it is important to maintain the momentum. In an information technology context, implementation encompasses all the processes involved in getting new software or hardware operating properly in its environment, including installation, configuration, running, testing, and making necessary changes.

## Implementation Steps

The implementation phase contains several steps some of them are: the installation of the production system, training of the initial users and the delivery of the documentation. In this section, we will describe how we implement our system and we will describe the system’s services with some samples of them.

## Implementation Procedure

Implementation phase is very important because to have a great system we must implement the system’s process in an efficient and effective way. At the beginning XMAPP tools was downloaded and installed, next the database and database tables were created, and the PHP codes to connect application to the database is written.

Developing application is started by installing the **android studio 3.2** and android library, then screens layout was created and was connected to the databases.

### Hardware requirements

1. Laptop I5core/ 4GB

This laptop was used to install and run needed program and tools, android studio needs at least core i5 microprocessor to run and at least 4GB of ram.

### Software requirements

1. Android Studio

Android Studio is an integrated development environment (IDE) for the Android platform. Android Studio provides integrated Android developer tools for development and debugging.

1. Android SDK

Android SDK (software development kit) provides a set of programming development tools that allows the programmer to develop certain application for a specific platform.

1. Windows 8.1

It is an operating system for the used laptop

## Main Codes

### Login code

This code used for check if the username and password in the database

﻿<?php

include 'conn.php';

$username = $\_POST ['username'];

$password = $\_POST ['password'];

$type = $\_POST ['type'];

if ($type == 'customer') {

$users = mysqli\_query($db, "Select \* from customer where username='$username' and password = '$password'");

echo mysqli\_error($db);

}

if ($type == 'farmer') {

$users = mysqli\_query($db, "Select \* from farmer where username='$username' and password = '$password'");

echo mysqli\_error($db);

}

if ($type == 'admin') {

$users = mysqli\_query($db, "Select \* from admin where username='$username' and password = '$password'");

echo mysqli\_error($db);

}

if ($type == 'employee') {

$users = mysqli\_query($db, "Select \* from employee where username='$username' and password = '$password'");

echo mysqli\_error($db);

}

$num = mysqli\_num\_rows($users);

if ($num == 0) {

echo '{"feed":[{"id":"-1"}]}';

} else {

while ($user = mysqli\_fetch\_array($users, MYSQLI\_NUM)) {

echo '{"feed":[{"id":"' . $user[0] . '"}] }';

}

}

?>

### Create Account Code

This code in android studio to connect to database and create user account

﻿<?php

// put your code here

include 'conn.php';

$name = $\_POST ['name'];

$mobile = $\_POST ['mobile'];

$email = $\_POST ['email'];

$city = $\_POST ['city'];

$dob = $\_POST ['dob'];

$username = $\_POST ['username'];

$password = $\_POST ['password'];

$type = $\_POST ['type'];

if ($type == 'customer') {

$result = mysqli\_query($db, "insert into customer values (null, '$name','$mobile','$email','$city','$dob','$username', '$password')");

echo mysqli\_error($db);

} else {

$result = mysqli\_query($db, "insert into farmer values (null, '$name','$mobile','$email','$city','$dob','$username', '$password')");

echo mysqli\_error($db);

}

if ($result) {

echo '{"feed":[{"id":"1"}]}';

} else {

echo '{"feed":[{"id":"-1"}]}';

}

?>

### Add Event Code

this code for connecting to database to add event

﻿<?php

// put your code here

include 'conn.php';

$title = $\_POST ['title'];

$details = $\_POST ['details'];

$type = $\_POST ['type'];

$image = $\_POST ['image'];

$date = date('Y-m-d H:i:s');

$img = rand(11111, 99999).".png";

file\_put\_contents('./' . $img, base64\_decode($image));

$root = 'http://' . $\_SERVER['HTTP\_HOST'] . '/';

$img = $root.$img;

mysqli\_query($db, "insert into new values (null, '$title','$details','$date','$type','$image')");

echo mysqli\_error($db);

echo '{"feed":[{"id":"1"}]}';

?>

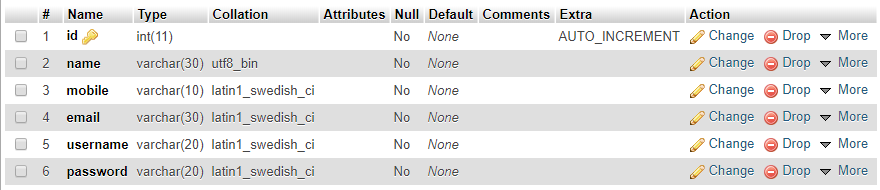
### View Cart Code

This code for view cart screen

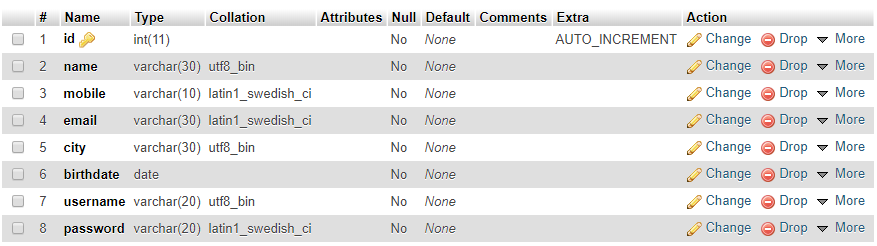
**package** com.example.dell.greengrow.Customer.ViewCart;  
  
**import** android.app.ProgressDialog;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.support.v7.widget.GridLayoutManager;  
**import** android.support.v7.widget.RecyclerView;  
**import** android.util.Log;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.Toast;  
  
**import** com.android.volley.DefaultRetryPolicy;  
**import** com.android.volley.Request;  
**import** com.android.volley.RequestQueue;  
**import** com.android.volley.Response;  
**import** com.android.volley.VolleyError;  
**import** com.android.volley.toolbox.StringRequest;  
**import** com.android.volley.toolbox.Volley;  
**import** com.example.dell.graangrow.R;  
**import** com.example.dell.greengrow.Session;  
  
**import** org.json.JSONArray;  
**import** org.json.JSONException;  
**import** org.json.JSONObject;  
  
**import** java.util.ArrayList;  
**import** java.util.HashMap;  
**import** java.util.Map;  
  
**public class** ViewCartActivity **extends** AppCompatActivity {  
  
 RecyclerView **recyclerView**;  
 Session **session**;  
 ProgressDialog **progressDialog**;  
 RequestQueue **queue**;  
 ArrayList<CartDetails> **myorderDetails**;  
 CartAdapter **adap**;  
 **private** GridLayoutManager **gridLayoutManager**;  
 Button **send**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_view\_cart***);  
 getSupportActionBar().setTitle(**"My Cart"**);  
  
 **recyclerView** = (RecyclerView)findViewById(R.id.***recycler\_view***);  
 **send** = (Button)findViewById(R.id.***send***);  
 **session** = **new** Session(getBaseContext());  
 **progressDialog** = **new** ProgressDialog(**this**,R.style.***AppTheme\_Dark\_Dialog***);  
 **queue** = Volley.*newRequestQueue*(ViewCartActivity.**this**);  
 **myorderDetails** = **new** ArrayList<>();  
 **adap** = **new** CartAdapter(getBaseContext(), **myorderDetails**,ViewCartActivity.**this**);  
 **gridLayoutManager** = **new** GridLayoutManager(**this**, 1);  
 **recyclerView**.setLayoutManager(**gridLayoutManager**);  
 **recyclerView**.setAdapter(**adap**);  
 getAllProduct();  
 **send**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **if**(**myorderDetails**.isEmpty()){  
 Toast.*makeText*(ViewCartActivity.**this**, **"Your Cart is Empty"**, Toast.***LENGTH\_SHORT***).show();  
 }**else** {  
 SendOrder();  
 }  
  
 }  
 });  
 }  
 **private void** getAllProduct(){  
 **progressDialog**.setMessage(**"Please Wait ..."**);  
 **progressDialog**.setCanceledOnTouchOutside(**false**);  
 **progressDialog**.setCancelable(**false**);  
 **progressDialog**.show();  
 StringRequest stringRequest = **new** StringRequest(Request.Method.***POST***,  
 **"http://ratebha1122-001-site1.ctempurl.com/cart.php"**, **new** Response.Listener<String>() {  
  
 @Override  
 **public void** onResponse(String response) {  
 JSONObject jsonResponse = **null**;  
 **try** {  
 jsonResponse = **new** JSONObject(response);  
 JSONArray result = jsonResponse.getJSONArray(**"feed"**);  
 **progressDialog**.dismiss();  
 **if** (result.length() == 0) {  
  
 finish();  
 Toast.*makeText*(getBaseContext(), **"No Product In Cart!"**, Toast.***LENGTH\_SHORT***).show();  
 }**else** {  
 **for** (**int** i = 0; i < result.length(); i++) {  
 JSONObject object = result.getJSONObject(i);  
 CartDetails data = **new** CartDetails();  
 data.setId(object.getString(**"id"**));  
 data.setProduct(object.getString(**"product"**));  
 data.setQuantity(object.getString(**"quantity"**));  
 data.setPrice(object.getString(**"price"**));  
 **myorderDetails**.add(data);  
 **adap**.notifyDataSetChanged();  
 }  
 }  
  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 **progressDialog**.dismiss();  
 }  
  
  
 Log.*d*(**"Response"**, response);  
 }  
 }, **new** Response.ErrorListener() {  
 @Override  
 **public void** onErrorResponse(VolleyError error) {  
 **progressDialog**.dismiss();  
 Toast.*makeText*(getBaseContext(), **"No Internet Connection !"**, Toast.***LENGTH\_SHORT***).show();  
  
 }  
 }) {  
  
 @Override  
 **protected** Map<String, String> getParams() {  
 Map<String, String> params = **new** HashMap<String, String>();  
 params.put(**"customerid"**, String.*valueOf*(**session**.getId()));  
 **return** params;  
 }  
 };  
 stringRequest.setRetryPolicy(**new** DefaultRetryPolicy(  
 30000,  
 DefaultRetryPolicy.***DEFAULT\_MAX\_RETRIES***,  
 DefaultRetryPolicy.***DEFAULT\_BACKOFF\_MULT***));  
 **queue**.add(stringRequest);  
  
 }  
 **private void** SendOrder(){  
 **progressDialog**.setMessage(**"Please Wait ..."**);  
 **progressDialog**.setCanceledOnTouchOutside(**false**);  
 **progressDialog**.setCancelable(**false**);  
 **progressDialog**.show();  
 StringRequest stringRequest = **new** StringRequest(Request.Method.***POST***,  
 **"http://ratebha1122-001-site1.ctempurl.com/sendorder.php"**, **new** Response.Listener<String>() {  
  
 @Override  
 **public void** onResponse(String response) {  
 Toast.*makeText*(ViewCartActivity.**this**, **"Order Sent"**, Toast.***LENGTH\_SHORT***).show();  
 **progressDialog**.dismiss();  
 finish();  
 Log.*d*(**"Response"**, response);  
 }  
 }, **new** Response.ErrorListener() {  
 @Override  
 **public void** onErrorResponse(VolleyError error) {  
 **progressDialog**.dismiss();  
 Toast.*makeText*(getBaseContext(), **"No Internet Connection !"**, Toast.***LENGTH\_SHORT***).show();  
  
 }  
 }) {  
  
 @Override  
 **protected** Map<String, String> getParams() {  
 Map<String, String> params = **new** HashMap<String, String>();  
 params.put(**"customerid"**, String.*valueOf*(**session**.getId()));  
 **return** params;  
 }  
 };  
 stringRequest.setRetryPolicy(**new** DefaultRetryPolicy(  
 30000,  
 DefaultRetryPolicy.***DEFAULT\_MAX\_RETRIES***,  
 DefaultRetryPolicy.***DEFAULT\_BACKOFF\_MULT***));  
 **queue**.add(stringRequest);  
  
 }  
}

## Database Tables

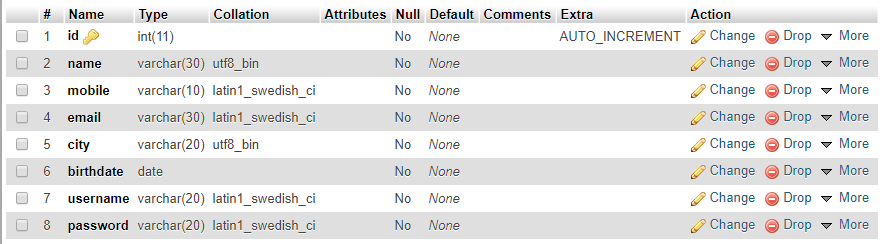
### Admin table



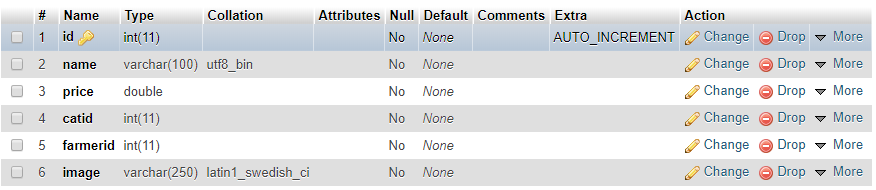
### Farmer table



### Cutomer table



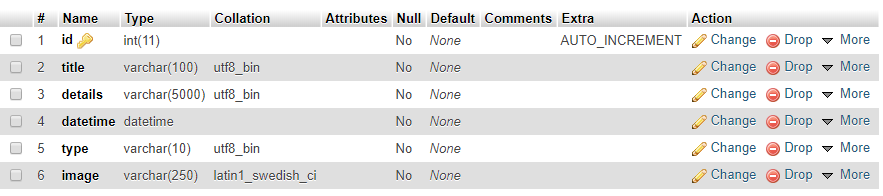
### Product table



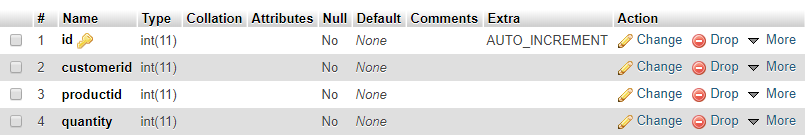
### Category Table



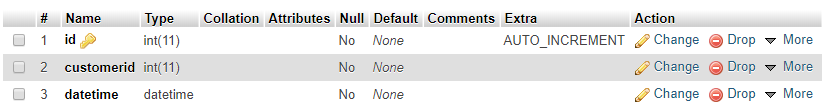
### Event/News Table



### Cart Table



### Oerder Table



## Screenshots

This suction contains the screenshots of the system interfaces, system interfaces created using android studio which allows developed to either create interfaces in XML programming language or be design mode.

The following is the application screens.



Figure ‎5.1 Splash screen

This is the splash screen of the application, this screen will be displayed for 3 seconds, then the application moves automatically to next screen.



Figure ‎5.2 Main screen

This is the main screen of the application, from this application the user can go to new account screen, login screen, about application screen.

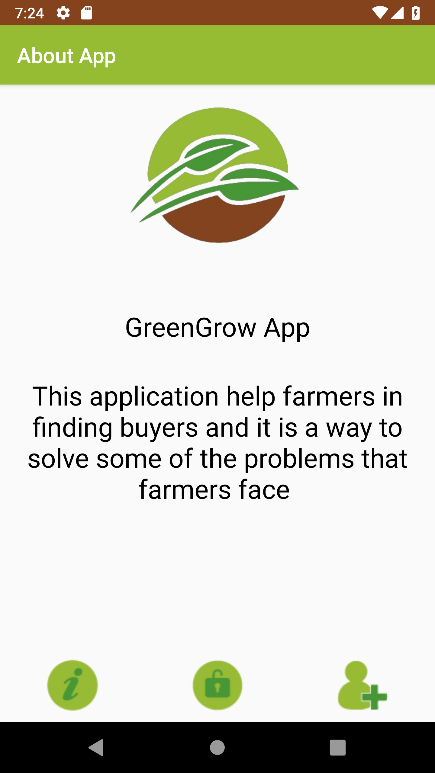


Figure ‎5.3 About screen

This screen shows information about the application.

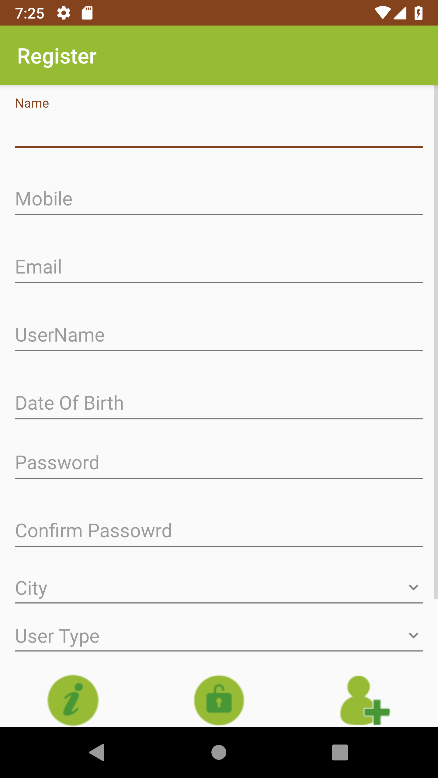


Figure ‎5.4 Registration screen

This screen allows the user to create new account, the user must enter his information and press register button.

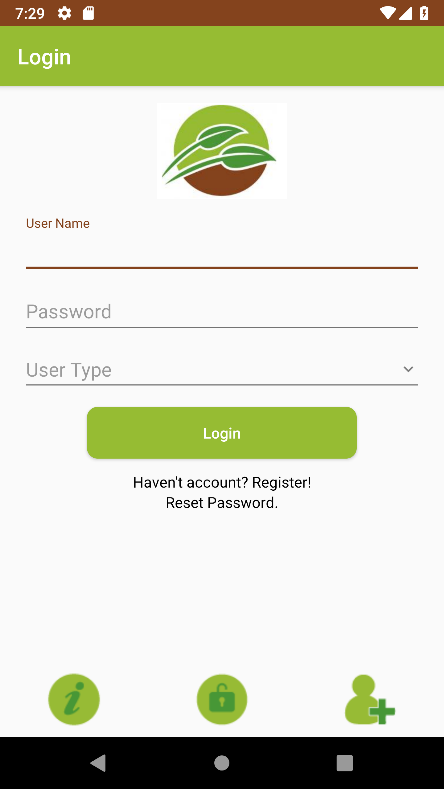


Figure ‎5.5 Login screen

This screen enables user to login to his account, the user enters his username and password and press login button, from this screen the user can enter the forget password screen or create account screen.

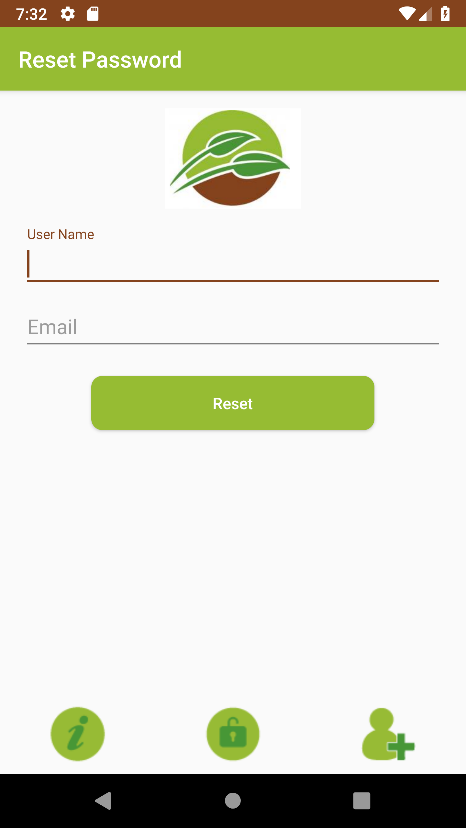


Figure ‎5.6 Reset password screen

This screen enables user to reset his password in case of forget a password, the user need to enter username and email.

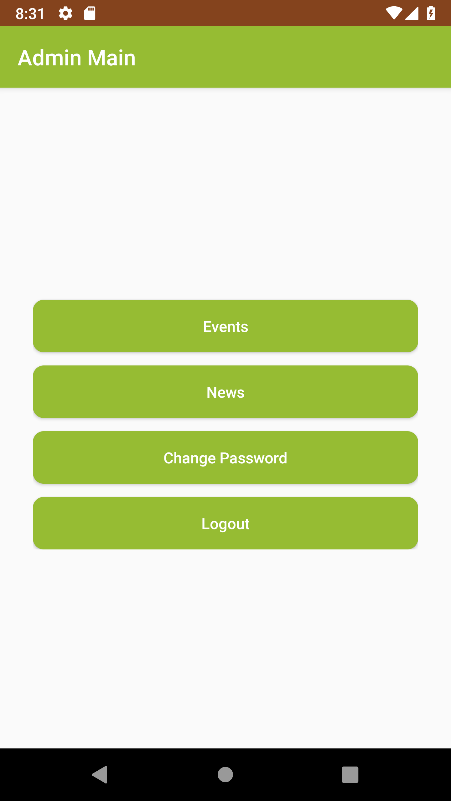


Figure ‎5.7 Admin main screen

This screen is the main screen of administrator, from the screen the admin can enters many screens, events screen, news screen, change password and logout,

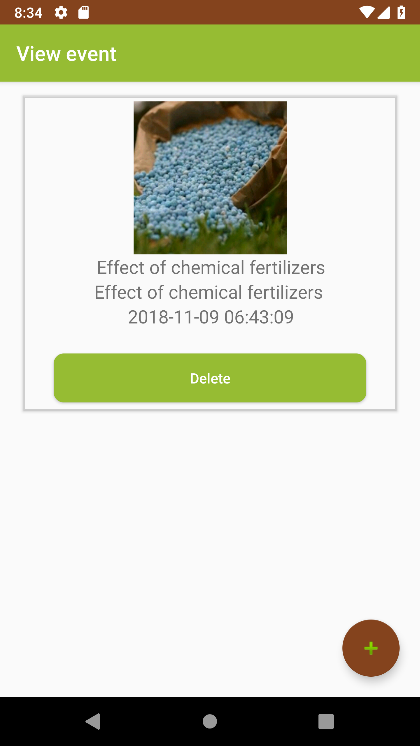


Figure ‎5.8 events screen

This screen shows all events to administrator, the administrator can delete event or add new event.

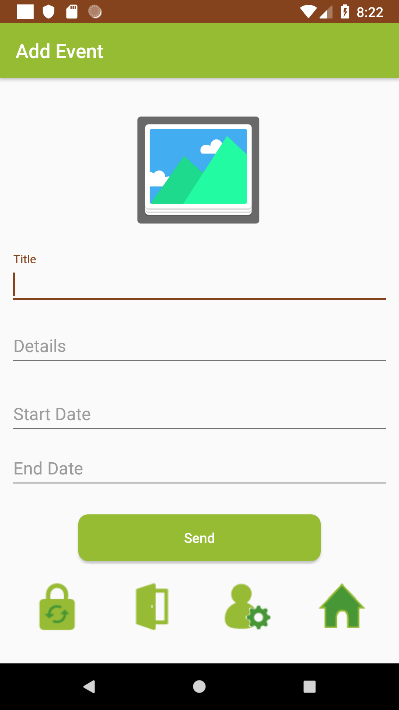


Figure ‎5.9 add event screen

This screen is for adding new event, the administrator must add title of event, description and select the image, then press send button.

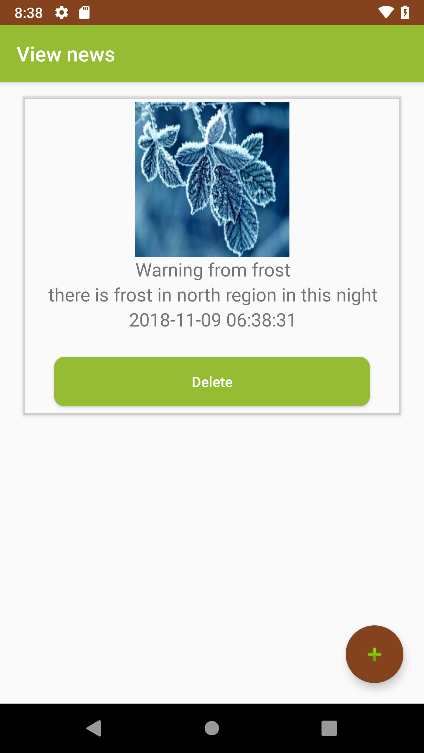


Figure ‎5.10 news screen

This screen shows all news to administrator, the admin can delete news or add new.



Figure ‎5.11 add news screen

This screen allows admin to add news, the admin need to select image and enters title and detail, then press send button.

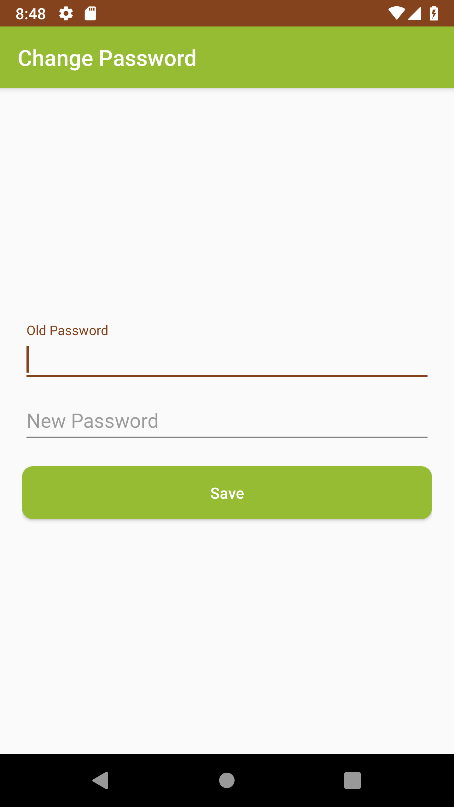


Figure ‎5.12 Change password screen

This screen allows the administrator to change his password, the admin need to enter current password and new password.

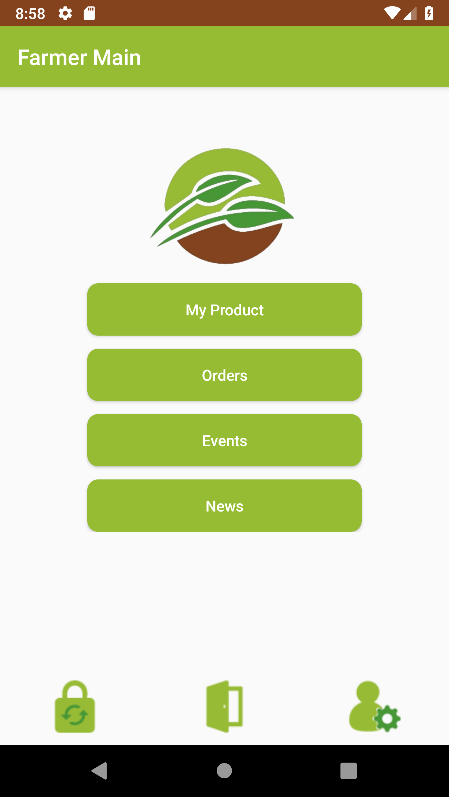


Figure ‎5.13 Farmer main screen

This screen is the main screen for farmer, from this screen the farmer can enters products screen, orders screen, events, news, change password, edit profile and the farmer can logout from account.

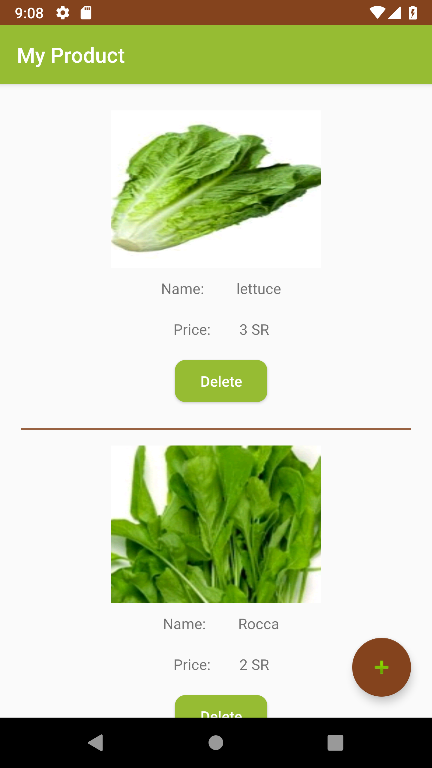


Figure ‎5.14 Products list screen

This screen shows the list of products, the farmer can delete any of the products or add new product.

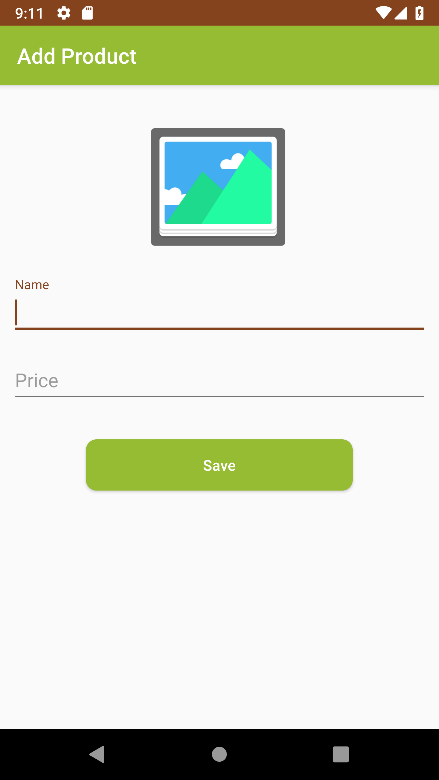


Figure ‎5.15 add new product

This screen allows the farmer to add new product, the farmer need to add name of product and price.

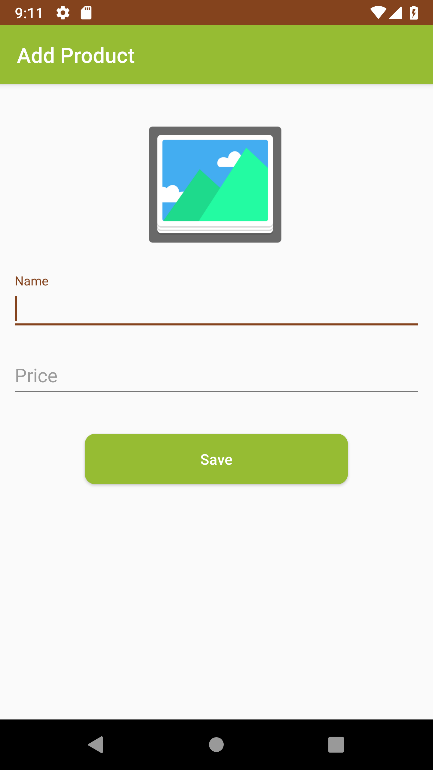


Figure ‎5.16 Events screen

This screen views all events, the farmer can press event to view the discussion about the event.

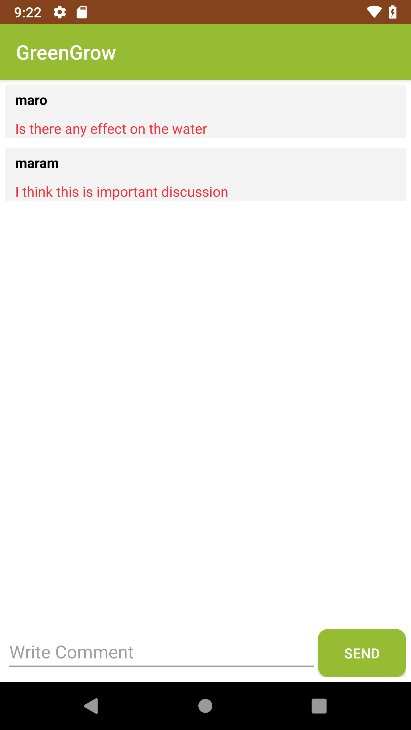


Figure ‎5.17 Event discussions

This screen shows discussion about even, the farmer can add new message and press send button.

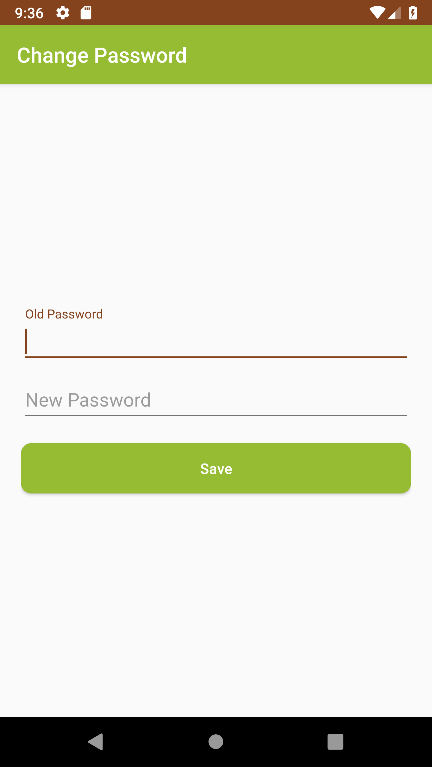


Figure ‎5.18 Change password screen

This screen allows farmer to change his password by entering the old and new password then press save button.

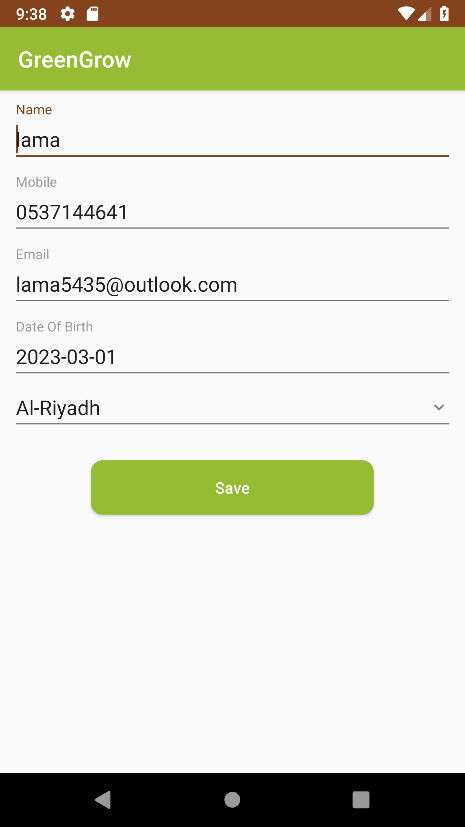


Figure ‎5.19 Edit profile

This screen allows the farmer to edit his profile.

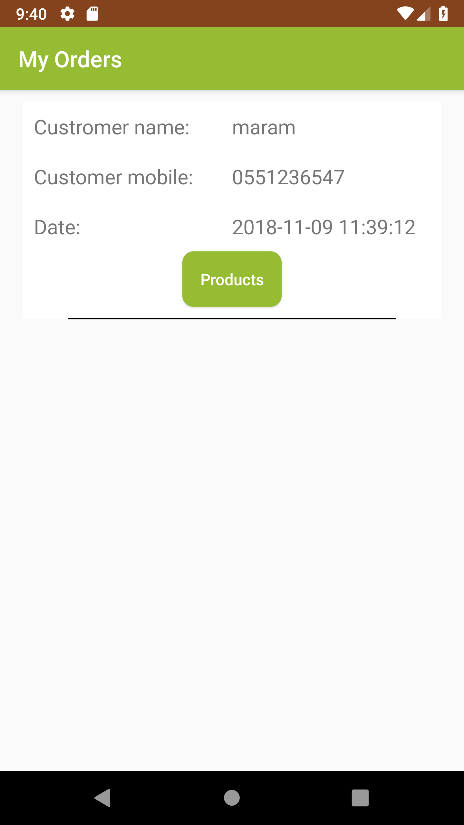


Figure ‎5.20 Order screen

This screen allows the farmer to view all orders, and customer information.

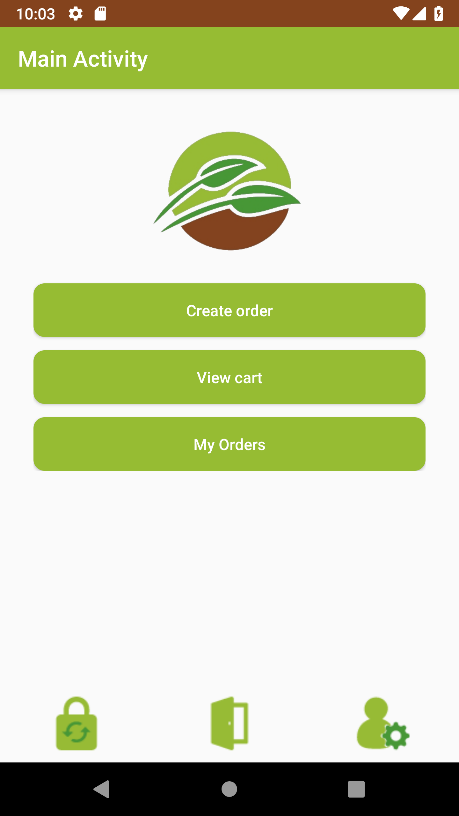


Figure ‎5.21 Customer main screen

This screen is the main of customer, the customer can create order, view cart, view order history, change password, and edit his profile.



Figure ‎5.22 categories screen

From this screen the customer can choose the category to view all product from that category.

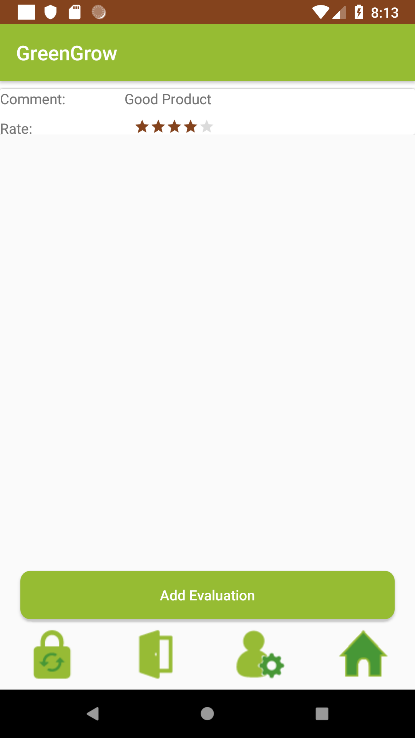


Figure ‎5.23 Product evaluations

This screen shows the customers evaluations of the products, the customer can view this evaluation before he add product to cart

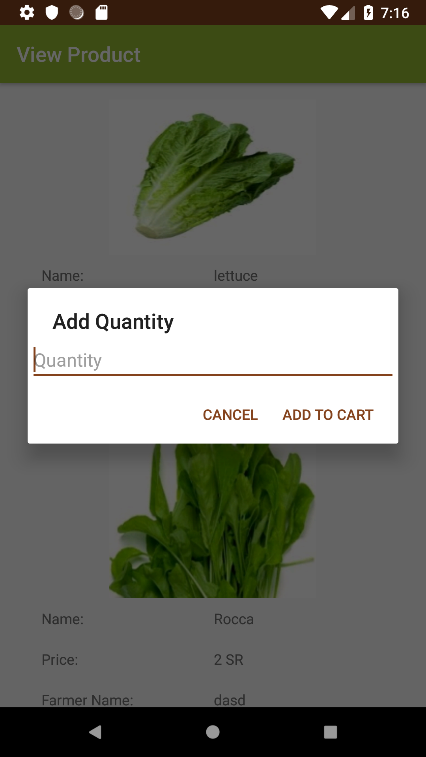


Figure ‎5.24 add product to cart screen

This screen views all product to customer, the customer can add any product to cart.

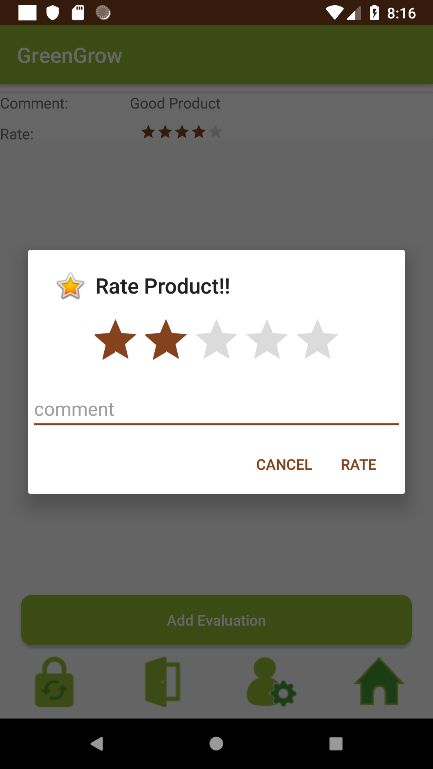


Figure ‎5.25 Add evaluation screen

This screen allows the customer to give an evaluation to products, he/she can write comment and rate the product from 1 to 5 stars.

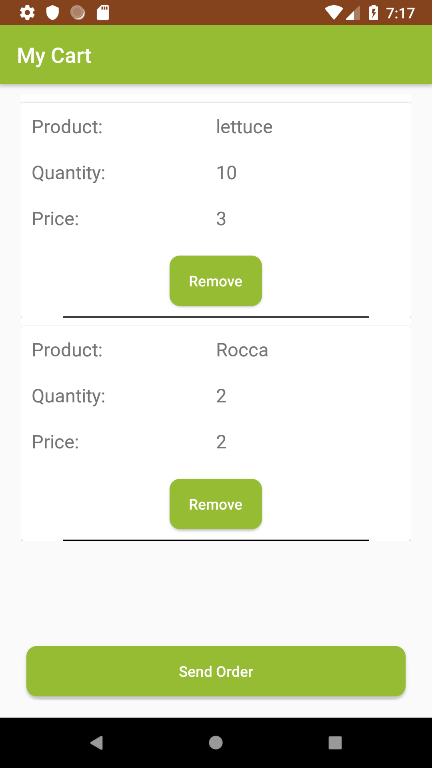


Figure ‎5.26 cart screen

This screen views all product in the cart and the quantity and price of each product, the customer can remove any of the product from cart, or send the order.

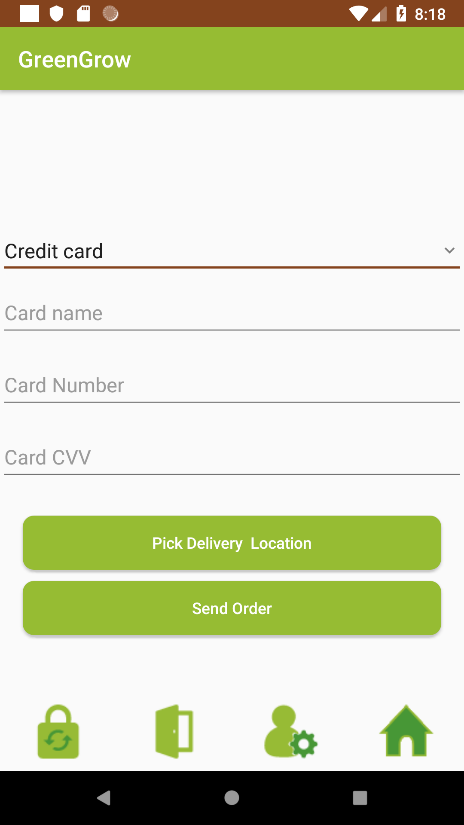


Figure ‎5.27 Payment screen

This screen allows the customer to select payment method and pick the delivery GPS location

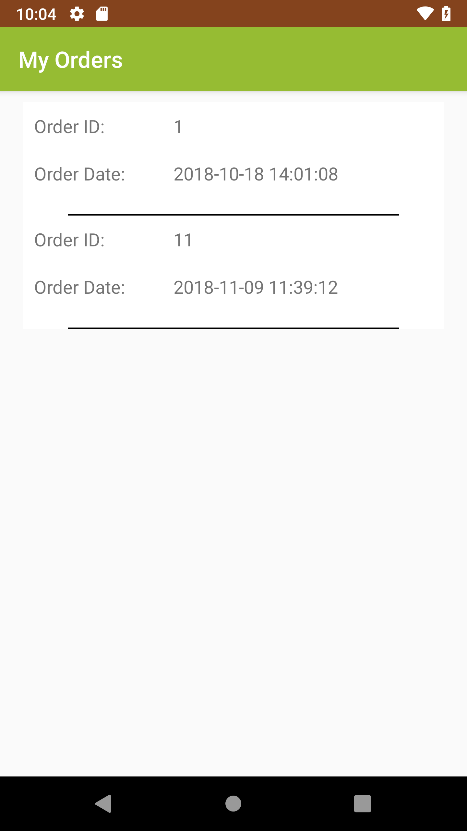


Figure ‎5.28 customer orders list

This screen shows the list of orders for customer.

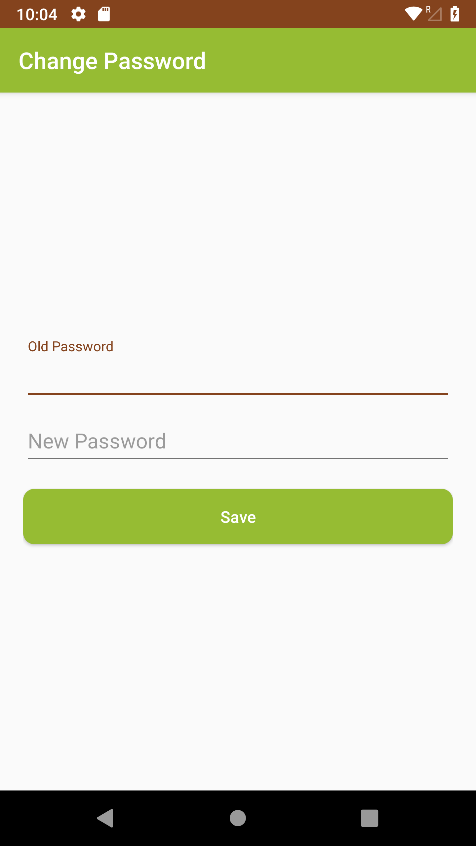


Figure ‎5.29 Change password screen

This screen allows the customer to change the password, he must enter old and new password.

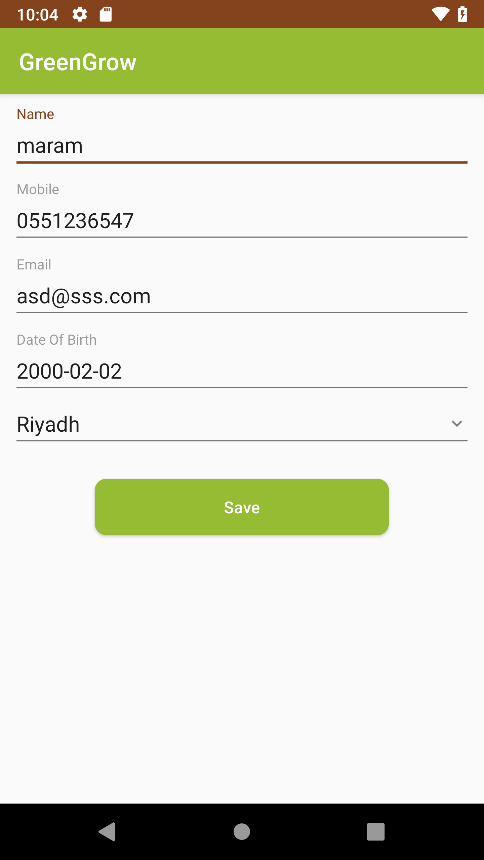


Figure ‎5.30 Edit profile screen

This screen allows the customer to edit his profile.

Chapter 6

SYSTEM TESTING

# CHAPTER 6: TESTING

## Introduction

This is a critical phase in any systems project. It is designed to find out whether a software system meets the acceptance criteria and determines if the system ready for use. User requirements play a basic role as they are usually subject to review and change. It is often the final phase of the project and involves a lot of the end users. It should be done in an environment that simulates the real world or production environment.

In this phase, most of the technical bugs might be fixed during various testing processes in project life cycle. The functionality and usability of the system must be tested.

## Testing types

There are several testing types and methods like the following:

### White box testing

It is a method of testing software that tests internal structures or workings of an application.

### Black box testing

Also known as Behavioral Testing, is a software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional.

### Functional requirements

Functional testing is a software testing process used within software development in which software is tested to ensure that it conforms with all requirements.

### Unit testing

Unit testing, a testing technique using which individual modules are tested to determine if there are any issues by the developer himself. It is concerned with functional correctness of the standalone modules.

### System testing

System testing is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system’s compliance with the specified requirements

## Testing Plan

We plan for testing our system, that is dividing the operation into three main parts

1. Admin side
2. Farmer side
3. Customer side

### Admin Functionalities

1. Login
2. Logout
3. Add news.
4. Add Events
5. Delete news
6. Delete event
7. View News
8. View Events
9. Change password.

### Farmer Functionalities

1. Login
2. Logout
3. Register
4. View products
5. Add product
6. Delete product
7. View orders
8. View orders products
9. View events
10. View event discussion
11. Write comment of event
12. View news
13. Edit profile
14. Change password

### Customer Functionalities

1. Login
2. Logout
3. Register
4. View products
5. Buying products
6. Edit profile
7. View orders
8. Change password

## Test Cases

### Admin Side

Table 6.1 shows the test cases for administrator side

Table ‎6.1 Admin side test cases

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | case | Objective | Input | Expected Results | Result |
| 1 | Login | Check admin login function | Username: admin  password: 123 | Login successfully if the username and password is correct and error message shown if the username or password is incorrect. | pass |
| 2 | View news | Check the news screen | - | The news list must be viewed to the administrator. | pass |
| 3 | Delete news | Check deleting function |  | The news must be deleted | pass |
| 4 | Add news | Check the function of adding news | News title, news details and image | The new must be added and appear in list | Pass |
| 5 | View events | Check the events screen | - | The events list must be viewed to the administrator. | pass |
| 6 | Delete event | Check deleting function |  | The event must be deleted | pass |
| 7 | Add event | Check the function of adding event | event title, event details and image | The event must be added and appear in list | Pass |
| 8 | Change password | Check the change password | Old pass:123  New pass: A12345678 | Password changed successfully | pass |
| 9 | Edit profile | Check the edit profile screen | Any user information | The new data saved | Pass |
| 10 | Logout | Check user logout function | - | The user must be logged out | pass |

### Farmer Side

Table 6.2 shows the test cases for farmer side.

Table ‎6.2 farmer side test cases

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | case | Objective | Input | Expected Results |  |
| 1 | Register | Check create account function | Information of user | Account created and can login | pass |
| 2 | Login | Check farmer login function | Username: lama21  password: lama321 | Login successfully if the username and password is correct and error message shown if the username or password is incorrect. | pass |
| 3 | Logout | Check farmer logout function | - | The user must be logged out | pass |
| 4 | View products | View orders | Check products screen | The product list must be shown | pass |
| 5 | Add product | Check the add product function | Enters product information | The product must be saved in database | pass |
| 6 | Delete product | Check deleting function |  | The product must be deleted | pass |
| 7 | View orders | Check orders screen | - | The orders list must be shown | pass |
| 8 | View orders products | Check view products of order function | - | The order products must be listed | Pass |
| 9 | View events | Check events screen | - | The events list must be shown | pass |
| 10 | View news | Check news screen | - | The news list must be shown | pass |
| 11 | View event discussion | Check the View event discussion function | - | The discussion must be changed | Pass |
| 12 | Edit profile | Check the editing function | Enter the profile page | The information if user must be shown and the user can change the information and save it. | pass |
| 13 | Change password | Check the change password | Old pass:321  New pass: A12345678 | Password changed successfully | pass |

### Customer side

Table 6.3 shows customer test cases

Table ‎6.3 Customer test cases

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | case | Objective | Input | Expected Results |  |
| 1 | Register | Check create account function | Information of user | Account created and can login | pass |
| 2 | Login | Check delivery login function | Username: lama21  password: lama321 | Login successfully if the username and password is correct and error message shown if the username or password is incorrect. | pass |
| 2 | Logout | Check customer logout function | - | The customer must be logged out | pass |
| 4 | View products | Check products screen | - | The product list must be shown | pass |
| 5 | Buying products | Check Buying product function | - | The function is worked fine | pass |
| 6 | Get bill | Check getting bill function | - | The customer must get bill after send order | Pass |
| 7 | Add product to cart | Check the adding product to cart function | - | The product must be added to cart | Pass |
| 8 | View products in cart | Check cart products screen | - | The cart product list must be shown | pass |
| 8 | Remove product from cart | Check remove product from cart | - | The product must be removed from cart | Pass |
| 10 | Create order | Check the create order function | Add products to cart | The order must be saved in database | pass |
| 11 | View orders | Check orders screen | - | The orders list must be shown | Pass |
| 12 | Edit profile | Check the editing function | Enter the profile page | The information if customer must be shown and the user can change the information and save it. | pass |
| 13 | Change password | Check the change password | Old pass: lama321  New pass: A12345678 | Password changed successfully | pass |

## Testing results

Table 6.4 shows the testing results of the application.

Table ‎6.4 Testing results

|  |  |  |  |
| --- | --- | --- | --- |
| No | Function | Success | Failed |
| 1 | Login | 100% | 0% |
| 2 | Logout | 100% | 0% |
| 3 | buying products | 100% | 0% |
| 4 | Getting bill | 100% | 0% |
| 5 | View products | 100% | 0% |
| 6 | Add product | 100% | 0% |
| 7 | Delete product | 100% | 0% |
| 8 | View news | 100% | 0% |
| 9 | Add news | 100% | 0% |
| 10 | Delete news | 100% | 0% |
| 11 | View event | 100% | 0% |
| 12 | Add event | 100% | 0% |
| 13 | Delete event | 100% | 0% |
| 14 | Register | 100% | 0% |
| 15 | Edit profile | 100% | 0% |
| 16 | Change password | 100% | 0% |
| 17 | logout | 100% | 0% |

## Results

After the testing stage was finished , all of system models and functions were worked fine and as expected.

Chapter 7

CONCLUSION

# CHAPTER 7: CONCLUSION AND FUTURE WORK

## Conclusion

In this project, an Android mobile application had been analyzed and designed to create an environment for customers and farmers where the farmer can sell his/her products directly to the customers.

This project had been followed the waterfall methodology, so it had been started from requirements gathering, then the requirements have been analyzed, and UML diagrams have been drawn for the proposed project.

Then design phase had been started by creating a database for the proposed system and user interfaces using Android studio.

The system had been developed for Android smartphones using android studio and MySQL database management system.

After the completion of the planning, analysis, design and project experience, the proposed project achieved all its goals.

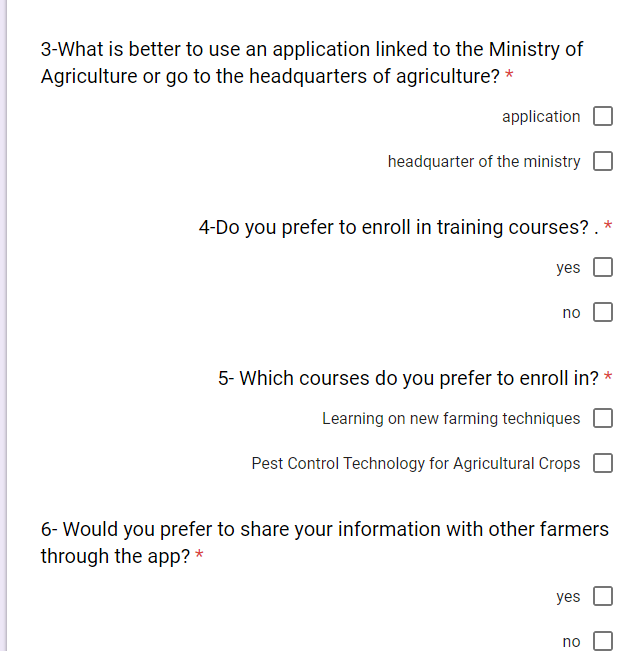
## Future works

1. Creating IOS edition of the application that support iPhone devices.
2. Adding notification features.
3. Publish the application to stores

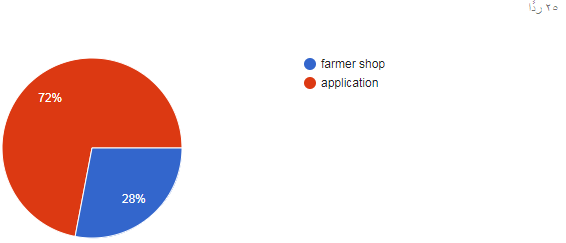
# Appendix (A)

## Questionnaire for agriculture

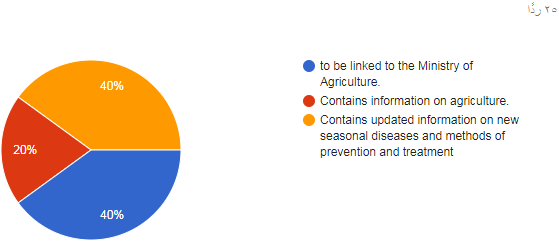
****



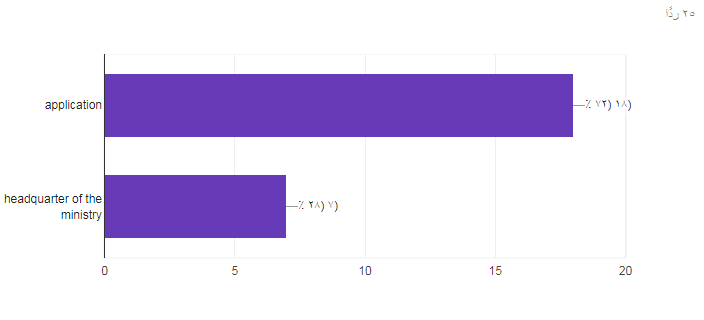
**1-What is the best purchase of the product from the application or from the shop where the farmer sells it?**



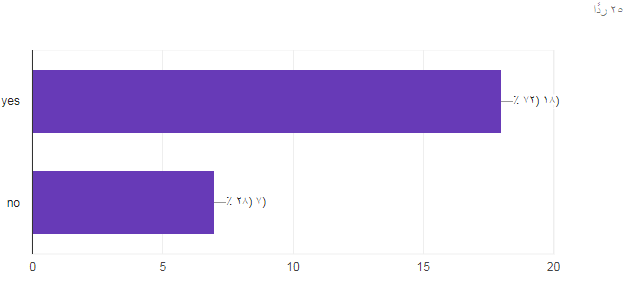
**2- What would you prefer to have in the app?**



**3- What is better to use an application linked to the ministry of agriculture or go to the headquarters of agriculture?**



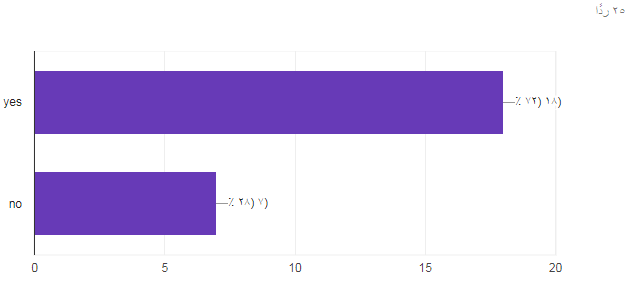
**4- Do you prefer to enroll in training courses?**



5**- Which courses do you prefer to enroll in?**



**6- Would you prefer to share your information with other farmers through the app?**



# References

[1]Roads and Maritime Services. (2017). Traffic signs. [online] Available at: http://www.rms.nsw.gov.au/roads/safetyrules/road-rules/signs.html [Accessed 9 Nov. 2017].

[2] UML 2 Toolkit, Hans-Erik Eriksson , Magnus Penker, Brian Lyons, and David Fade

[2] Systems Analysis and Design with UML, David Tegarden, Alan Denis, and Barbara Haley Wixom.

[3] The Unified Modeling Language User Guide (2nd Edition) Grady Booch, James Rumbaugh, and Ivar Jacobson, Addison-Wesley Professional

[4] McLaughlin, Brett PHP and MySQL: The Missing Manual 2011

[5] John M Wargo, phone gap Essentials: Building Cross-Platform Mobile Apps “2012

[6]plantix | Smart Crop Assistant", Plantix.net, 2018. [Online]. Available: https://plantix.net/ar. [accessed: 11- Apr- 2018].

[7]Available: http://agri-science-reference.blogspot.com/p/blog-page\_29.html2-. [Accessed: 11- Apr- 2018].

[8] Available: http://agri-science-reference.blogspot.com/p/blog-page\_29.html2-. [Accessed: 11- Apr- 2018].

[9] Don Fowley, Systems Analysis and Design, 5th Edition, 2012.

[10] Umsl.edu. (2018). [online] Available at: http://www.umsl.edu/~sauterv/analysis/presentations/w08/Methodology%20Info.doc [Accessed 14 Apr. 2018].

[11] [4]Bing.com, 2018. [Online]. Available: [http://www.bing.com/cr?IG=04C75C4A41C942979CEA3777A4EEA2D3&CID=06FA2888A1AC66B229452359A0036722&rd=1&h=Nk4XGl5N6hIMnx926hNHsv6RWFJdKR5ruNXbCxqaaw&v=1&r=http%3a%2f%2fwww.umsl.edu%2f%7esauterv%2fanalysis%2fpresentations%2fw08%2fMethodology%2520Info.doc&p=DevEx,5068.1](http://www.bing.com/cr?IG=04C75C4A41C942979CEA3777A4EEA2D3&CID=06FA2888A1AC66B229452359A0036722&rd=1&h=Nk4XGl5N6hIMnx926hNHsv6RWFJdKZR5ruNXbCxqaaw&v=1&r=http%3a%2f%2fwww.umsl.edu%2f%7esauterv%2fanalysis%2fpresentations%2fw08%2fMethodology%2520Info.doc&p=DevEx,5068.1). [Accessed: 14- Apr- 2018].