

Senior Design Project Report

CSE/EEE/ETE 499

Online Urban Farming: Introducing online platform for garden lovers



Submitted By
1511104042 Mubina Rahaman Jerin
1420477042 Fatiha Jahan

Supervisor
Mohammad Rezaul Islam– IMr
Senior Lecturer

ELECTRICAL AND COMPUTER ENGINEERING
NORTH SOUTH UNIVERSITY
Fall 2020]

Agreement Form

We take great pleasure in submitting our senior design project report on “Online Urban Farming: Introducing online platform for garden lovers”. This report is prepared as a requirement of the Capstone Design Project CSE/EEE/ETE 499 A & B which is a two semester long senior design course. This course involves multidisciplinary teams of students who build and test custom designed systems, components or engineering processes. We would like to request you to accept this report as a partial fulfillment of Bachelor of Science degree under Electrical and Computer Engineering Department of North South University.

Declared By:

.....
Name: Mubina Rhaman Jerin

ID: 1511104042

.....
Name: Fatiha Jahan

ID: 1420477042

Approved By:

.....
Supervisor
Mohammad Rezaul Islam
Sr. Lecturer, Department of Electrical and Computer Engineering
North South University, Dhaka, Bangladesh

.....
Dr. Rezaul Bari
Chair, Department of Electrical and Computer Engineering
North South University, Dhaka, Bangladesh

Abstract

Urban agriculture is the one of the most relevant environmental innovation of the modern era. Sustaining green habitation in city areas has a great social, environmental and commercial effect among the city dwellers. In fact, the decision of urban plantation and rooftop gardening in city can make overpopulated city area more suitable for living with greener environment. Implementation of innovative ideas in this domain can increase the chance of a healthy environment. IT products can certainly be a sustainable solution to network people in the noble cause to make their populated cities with more green habitation. Moreover, online software solution can ease the process of urban plantation in urban spaces like rooftops, balcony, enclosed spaces with little sunlight. More specifically acquiring the tools and services related to gardening is seldom in the city. Online software based products can help people to network to other people who have implemented knowledge about urban farming. Also, people can get to buy and sell agricultural tools as well as the harvest. In this project, we aim to design, implement and test a software based solution for buying, selling, networking and giving knowledge about urban agricultural products among the online users.

Table of Contents

	Chapter 1: Introduction	Page
1.1	Project Details	1
1.2	Background and Motivation	1
1.3	Project Goal	1
	Chapter 2: Technical Details	
2.1	Existing Solution	2
2.2	Proposed Solution	2
2.3	Solution Assessment	2
2.4	Design Alternative	4
2.5	Technical Design: Module Level	8
2.6	Technical Design: System Level	10
2.7	Required Skill	11
		14
	Chapter 3: Essential parts and Devices	18
3.1	Description of Components	25
3.2	Test Equipment	28
		30
	Chapter 4: Working Sheets	34
4.1	Work Breakdown Structure	45
4.2	Financial Plan and Costs	48
		55
	Chapter 5: Project Summary	56
5.1	Result and Discussion	57
5.2	Feasibility Study	61
5.3	Problem Faced and Solutions	65
5.4	Future Development	77
5.5	Conclusion	78
5.6	Project Demonstration Review	80
5.7	Brochure	81
5.8	Poster	86
5.9	IEEE format paper	92
	Appendix	110
A	Reference	111
B	Datasheet	112
C	Codes	
D	Survey Questionnaire	

List of Tables:

	Name	Page
1	Table of comparisons between existing solution and our solution	5
	Table of comparisons between First prototype and our Final Prototype	34
	List of modules of the First Prototype	48
	Task activity	55
	Financial cost diagram	60

List of Figures:

	Name	Page
1	Nogor Krish Home Page	3
2	FB market place of Prakiti Farming	7
3	Member registration of prakriti farming	10
4	Blog of Prakriti Farming	11
5	E-commerce Website of Ugaoo.com	12
6	Shopping Cart of Ugaoo.com	13
7	Smart Gardener website	14
8	Smart garden Blog	17
9	Smart cultiva website	34
10	Smart cultiva website(Page 2	35
11	System of Transaction	37
12	Actors and Agents of the Urban Farming Application	38
13	Graphical view of placing order	39
14	Provider creates a service module	40
15	Use case diagram	41
16	Top-Down View of the Application	43
17	Application Layer	42
18	Home Page of Urban Faming Application	43
19	Blog Page of Urban Faming Application	45
20	Authentication Page of Urban Faming Application	46
21	Custom Validation of Urban Faming Application	48
22	Custom Validation of Urban Faming Application(page 2	50
23	Nursery Portal of Urban Faming Application	61
24	Nursery Portal of Urban Faming Application	66
25	Response regarding services of Urban Faming Application	67
26	Response regarding service cost of Urban Faming Application	68
27	Response regarding Plant type choice of Urban Faming Application	69
28	Registration and login page of Mobile Application	71
28	Blog View pages of Mobile Application	77
30	User response on services	78
31	User response on User Friendly	81
32	Code Flow Urban Faming Application	83

CHAPTER 1

INTRODUCTION

1.1 Project Details:

In the modern era of science and technology, buying and selling of goods in ecommerce application became increasingly popular. Nowadays, almost everything is possible to get via online. In the context of Bangladesh, agriculture is the one of the most crucial sector for the growth of our middle income economy. Most of the developed cities in our country are densely populated. Even most of the cities are suffering rapid unplanned urbanization. This cause destruction of green habitation inside the city area. But green habitation is very important for maintaining a healthy air inside the city. Due to organizational factors, often city areas are deemed to have more commercial values and it caused no green habitation. In the context of Dhaka, roughly 3% of the area is covered by green habitation. It is a very low ratio of green habitation in terms of land area when we compare it with most developed cities all around the world.

Urban gardening or urban farming could be the perfect solution for reducing the overwhelming pollution inside Dhaka city. Urban farming is the cultivation of food within metropolitan cores as opposed to that in more pre-urban and rural areas [5,7,4]. Brooklyn Grange is the world's leading soil rooftop farming company and sustainability center for providing online services to its customers a online platform to get knowledge about urban farming and services of plantation confirmation on urban gardening. Bos presented the impact of social media for creating urban farming as a sustainable food network in city areas. He also pointed out the facts that can influence the development of online services for making urban farming easier concept to reach and cultivate urban household [2]. Urban farming can be a game changing solution for the developing green Dhaka city.

Online platform for ecommerce of agricultural products is really crucial for developing interest in urban farming among the city dwellers. Using the ecommerce site users can buy and sell garden products and also users can gain knowledge via sharing the tips regarding the gardening. Furthermore, logistics services via online platform can benefit the sales of the nursery owners who have little outreach of the customers. Online services of the ecommerce solution of the urban farming can consists usually delivering the gardening tools, seeds, manpower, 3D plans of the gardening, tips, and experimental designs. To power these online services our projects aims on developing both a web based and mobile based solution to help the users. In this project, we aim to develop an online platform where people can buy and sell their urban farming products as well as can acquire gardening services like gardening setup, loan decorating, balcony decorating from the different trained professionals. We also aim to develop a small android app in terms of an easy interface to use. This apps provides information about the latest agriculture advice, various farming tips, nursery trees prices, nursery stores and green lover's interactions platform and green lovers can easily take help of agriculture experts using this app.

1.2 Background and Motivation:

Urban agriculture (UA) is blooming around cities of the developed world as a response to the increasing urban population, the growing environmental awareness of the industrial food system and the need of addressing social gaps. These new local food systems aims to develop sustainable pathways that re-establish the relations between producers and consumers while boosting local economies and minimizing food-miles [8].

Although the Dhaka city was well known for its green resources (e.g. parks, open spaces, tree covers, forested areas etc.) but due to rapid rural-urban migration, unplanned urbanization, commercial development and industrialization the city is losing its greeneries at an alarming rate. Urban green spaces or greeneries are predominantly crucial for proper functioning of the ecosystem in any urban environment [2]. Urban green spaces can play critical role in conserving biodiversity, protecting water resources, improving microclimate, sequestering carbon, and even supplying a portion of the fresh food consumed by urban dwellers [7]. Author in [1] asserted that the green resources in Dhaka are overwhelmed by a number of limitation inherent to the exceptional mode of urban development, rapid increase of urban population, transformation of green and other open spaces into other types of land use. Lack of proper planning and implementation and management restrictions also hamper green resources in Urban areas [3].

Socially acceptable urban agriculture businesses: this paper is suggesting that we can take this urban rooftop farming as business purpose [9]. In a study, sweet potato was cultured with a lightweight hydroponic system on a rooftop and performances of the plants to

reduce excess temperature rise & to yield tuberous roots were examined in a summer season in Osaka, Japan [6]. Authors analyzed the importance of the framework for urban farming activities to be implemented within cities environment [10]. Considering above facts, urban farming is an important subject matter to the development of a healthy Dhaka City. To spread the urban farming among the mass people we need a sustainable software product to leverage the power of social media networking and ecommerce popularity of this era.

Urban farming can be a game changing solution for the developing green Dhaka city. But most of the people barely know about urban gardening and urban farming rules. Furthermore, they don't have enough resources to make urban farms in rooftop areas. Most of the city dwellers are not interested in urban gardening due to lack of soil materials, seeds etc. Also, city dwellers have lack knowledge of nursing process of trees in urban climate. On the other hand, nursing in Dhaka is operationalized via various privately owned nurseries. They sell seeds and plants to their customers so that city dwellers can build their own piece of urban garden. There is no online platform currently available for nursery owners to provide their plantation service to the consumers who want to grow gardens in their rooftop areas. Brooklyn Grange is the world's leading soil rooftop farming company and sustainability center for providing services to its customers a platform to get knowledge about urban farming and services of plantation confirmation on urban gardening. But the online service of the company can't be used in Bangladesh. The importance of local online services is really crucial for spreading the urban farming in Bangladesh.

Nagar Krishi is a local online platform for delivering gardening materials like seeds, soil, gardening tools. The software solution is web based and offers range of gardening materials although they don't offer 3rd party sales of gardening materials. The disadvantages of such online platform are lack of areas to expand. If the local nursery owners are included in the online platform to sell their gardening products to the customers it would be easier to sell their products to the customers.

Prakiti farming is another service for urban farming. They offered a website where user can register and be part of the organic food producer. The company provides necessary logistics to help urban farmers to grow their business. The advantages of the product are the ability to integrate customers into the urban farming business, the social awareness blog about urban farming etc. But the main disadvantages of urban farming include the lack of online services for the new gardeners of the system. They only can register. Other than they can't manage anything of the farm digitally.

Ugao is a complete software product with wide range of online service to its customers. Customers can buy different gardening kits, seeds, plants, customers can pay the price of the order via online payment gateways, and users can track their order. But the main drawbacks of the platform are no integration of 3rd party sellers, no management information option for the gardeners.

SmartGardener is a mobile application for creating urban garden using 3D tools. The application provides wide range of graphical tools for creating a 3D space of the urban garden. User can manage the garden and get important cultivation information based on the plantation provided by the user in 3D space. The main disadvantage of the application is there is no online service for buying and selling gardening materials.

Smartcultiva is a company in urban farming business who delivers IOT based technologies to the rooftop based farmers. Urban farmers can get different types of sensors to measure weather conditions for various kinds of plants inside the rooftop garden. The tools offered by the company helps the farms to monitor and analyze the crop harvested in the urban spaces. But they don't provide any type of gardening materials to the famers via their online service and this is the one of the main disadvantages of the company. On the other hand, the technology helps the urban farms to grow efficiently and farmers can get maximum output of their harvest using the tools provided by the company.

After reviewing many online platforms, it's really clear that online ecommerce service for growing urban faming is the perfect software based solution to prosper the green habitation in the cities of Bangladesh. We propose an online platform for giving services like rooftop plantation, rooftop beautifications, vertical gardening etc. The software is a B2B model for both plantation provider and consumer where the web application is the base platform of interaction of users. The web application contains a three separate user interaction system for providing the services for plantation provider, consumers and administrator of the software.

1.3 Project Goal:

In this project, our main aim to develop an online buying and selling platform of gardening material and harvests via 3rd party integration. 3rd parties include the nursery owners who wants to sell their gardening products, the individual customers who wants to sell their gardening harvests.

Online gardening platform for urban gardening can be implemented both as web application and mobile application. At first we modeled the activity diagram for our solution at the first stage of our research on the software development of the proposed product. Now we forwarded to implement the system design using the web application framework Laravel 5. Also we have to implement our software design into the mobile application development OS android.

In our product design approaches, we implemented the incremental development approach software design pattern. In this design, we implemented and test a sub-part of the whole system. Then analyze the feedback of the test to better the development of the next sub-part. Eventually, the whole system is increasingly developed via through testing of the implementation.

In our project, we implemented Laravel 5.2 framework for the development of our web application. Our web application consist three basic module for user interactions: User module, Producer module and the administrator module. Each of the module has different subparts which are very crucial each of the module. For Example, user module have authentication system, blogging tool, buying tool, garden showcase module, order a gardening tool and tracking of order. Producer module have more logistic features like

monitoring the sales, inventory management, order statistics etc. Admin module contains the necessary tools like blog management, rating management etc. features.

In short, the project aims to develop an E-commerce solution for buying selling products related to urban farming. In addition, the blog will be developed to increase social awareness via interaction among online users.

In our project, we have build an android apps and website .Through this app and website, we will build up a relationship between Nursery owner and Green lovers. Not only that, we have a feature where people can sell their own fresh vegetables. Our main goal is to build up a communication between customer and nursery owner. This is a special additional feature.

Let's discuss about the available features in our Web based app:

1. When someone enters the first page, it will be shown if he/she wants to register or sign in . If he is new to this app, he will choose the 'sign in' option .Then it will be shown 'Sign in' as customer or nursery owner . Choosing as customer , he will be logged in .After Opening an account ,he will be able to use our features.

Features:

Basic needs:

If you're considering a rooftop garden, there are a couple of directions you could go in. Fully planted green roofs, where the roof is covered with soil and the plants are in the ground, make great environmental sense, but they are too difficult for homeowners to undertake on their own. Green roofs can easily top 100 lbs. per sq. ft., before adding people. You would need to hire a structural engineer or architect to conduct a

structural analysis and probably a professional company to install it. In this option ,There will be instructions for customer .

Model:

There will be various types of design with pictures. This design will be available for our clients. We will offer them as per budget. Some will be costly, some will be cheaper.

Nursery owner:

Nursery owner can promote their nursery through our apps. They can upload the nursery's picture, what type of trees, flowers, garden accessories are available there.

Call:

After choosing the nursery owner, customer can call them.

Suppressing Diseases and Insects:

Types of Diseases

-  Apple scab
-  Apricot Bacterial canker
-  Blackleg
-  Blossom wilt
-  Botrytis
-  Brown Rot
-  Clematis wilt
-  Downy mildew
-  Fungal disease
-  Honey fungus
-  Greenback

- Fungal leaf spot

- Leaf curl

- Leek rust

Types of Pests:

- Ants, Bees and Wasps. Wasps, bees and ants (collectively known as hymenopterods) are one of the largest orders of insects. ...

- Bed Bugs. Bedbugs are small, elusive, and parasitic organisms all belonging to a family of insects called Cimicidae.

- Fleas.

- Flies.

- Myriapods

- Rodents.

- Snails

- Woodlice

- Wood Lovers(susa)

Organic Agriculture:

- The idea of organic agriculture

- Fertilizer

- Pheromone trap

- Organic fertilizer

- Modern Agriculture

- Green pesticides

- Fruit bagging

- Organic system
- Alternative agriculture
- Glue trap

Subscription

Customer can subscribe our package .There is there option.

- 1.Monthly based
- 2.Our own people
- 3.expert help

Blogger

This is a online journal where people can share their views on garden. Through this blog, people will know each other & able to share their feelings.

Videos

In this option, garden related videos will be available.

Our Mobile app project goal:

- Our mobile apps goal is to getting known by people. This mobile app will be a platform where green lovers and nursery owners can interact with each other.
- We want to build up an apps which can save people's time or reduce the workload that needs to be done or reducing complexity in a system that includes the number of active users.
- This apps will be an online community where green lovers and nursery owners can communicate with each other.
- In our apps, nursery owners can advertise their own nursery. Green lovers can buy nursery trees from here, app will update recent information about nurseries.

- Create employment opportunities for other people.
- Apps will be a marketplace where people can purchase.

CHAPTER 2

TECHNICAL DESIGN

2.1 Existing Solution:

Existing Web solutions

In this section, we discuss more thoroughly about some of the technology related to urban farming that presented at section 1.2 and give more graphical demonstration of the technologies. Each technologies describe in the previous sub-section have some specific feature for the urban farming business. In the table. 1 we shown the most relevant technologies related to urban faming business is shown. In the table the features provided by the each technologies.

Website/Technology	Category of Business	Features/Available products
http://nagarkrishi.com/	Retail sale of own garden kits	Shopping of garden kits
https://www.prakritifarsming.com/	Garden services	Membership based service for garden management
https://www.ugahoo.com/	Ecommerce	Garden services, online shopping of garden kits
https://www.smartgardener.com/	3D garden designing	3D design of garden space.
https://www.smartcultiva.com/	Sale of IOT devices for urban farming	Smart sensors of urban farming

Table 1: Table of existing solution

In the table we mentioned Nagarkrishi [11] where user can buy garden kits via calling in cell phone and facebook page. They don't offer any ecommerce service for the customers who wants to buy garden kits like seeds, soil, garden tools etc. They offer different types of garden management via the FB digital marketing. Most of the services offered by the company FB based. The Figure 1. graphical representation of their website.



on balconies and in houses, on walls and in offices. on rooftops and in classrooms. a new, green city.



want to be an urban farmer?

lets build your farm

we will teach you which plants to grow, how to take care of them and how to make the best use of your space. we will also look through



want us to farm for you?

have your own farm without any work

we will build your farm, plant what you want to eat, come as often as you like to look after



want to green your house or office?

make your space into an urban oasis

let us transform your building into a green space, and reduce your energy bills by



Figure 1: Nogor Krishi

Basically they get the orders from the clients via phone calls and then they give their services. They offer different website page to showcase their services. They also have a blog named “green talk” about different plant issues for the urban farmers. They integrated a FB chatbot extension to their web application so that customer can chat with the agents of Nagarkrishi. They do their online sales via facebook page. In short, most of the detailed technological ideas offered by Nagarkrishi is mostly based on the website they offered and many services are actually manual instead of own automated services.

Prakriti Farming [12] is another product related to urban agriculture. This company offers membership of gardening to its online users. Users who have urban spaces can register there and can grow their plants via the help of the company. The company provides labor and raw material support for the gardening. After that they sell the products of the urban farmer via the facebook page of the PrakritiFarming. The website also offers different articles of the customers who want to write about their experiences with urban gardening.

The blog option, member registration and fb marketplace of the product is presented in figure.

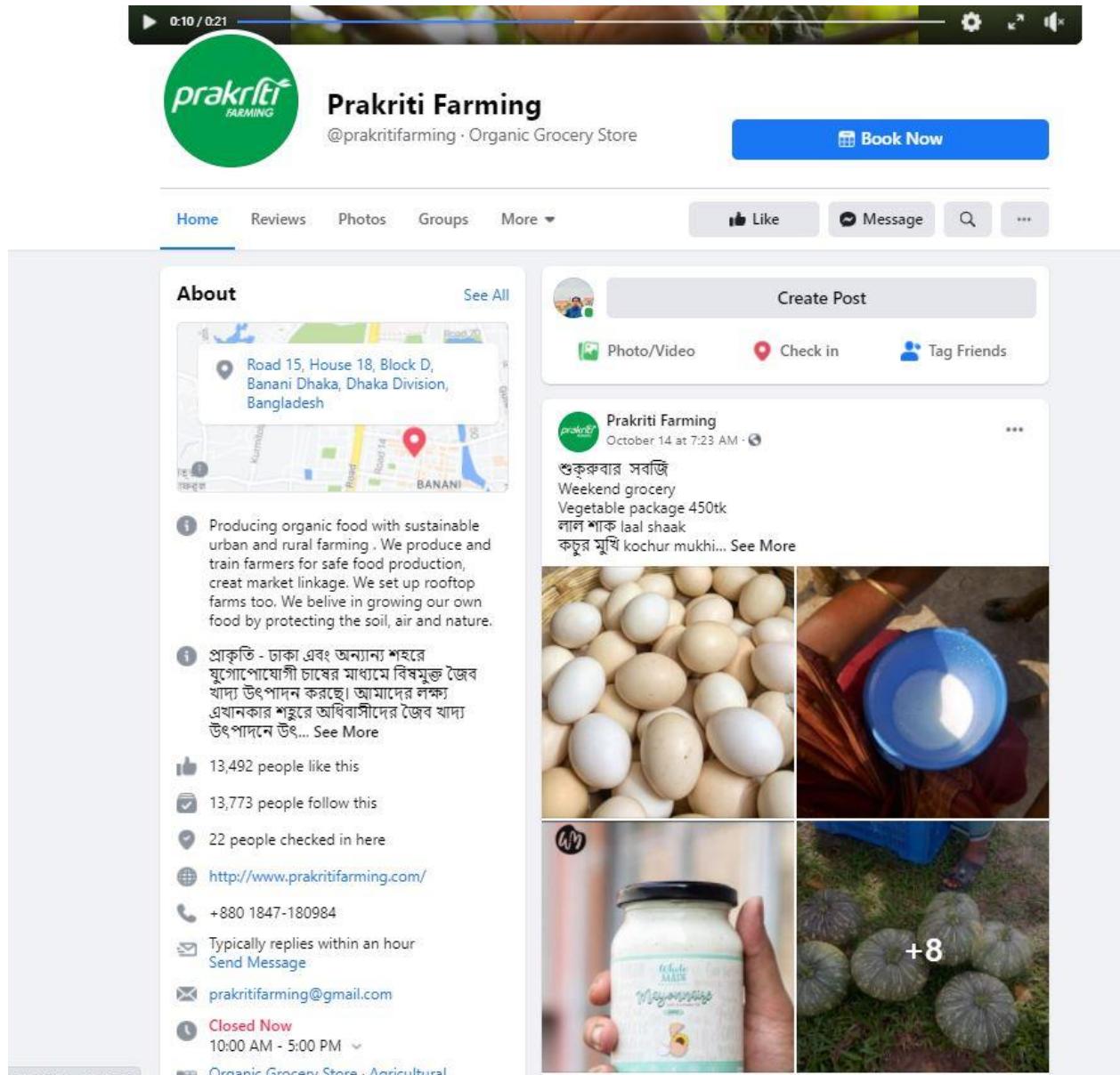


Figure 2: FB market place

The FB marketplace of the product offers all the necessary tools for order an service for the customers and also using the fb portal the company's customer can communicate with

customers for compliance issues. The core technology of the product is the idea behind the urban farming and also the membership registration of the gardeners who wants to

The screenshot shows the Prakriti Farming website's membership registration form. At the top, there is a navigation bar with links for About, Membership, Events, Support us, Contact, Resources, JOIN THE COMMUNITY, and a search icon. The main title "Membership Registration" is centered above the form fields. A note says "Fields marked with an * are required". The first field is "Select a package" with a dropdown menu showing "Beginner (2 months)". Below it is a "Full Name *" input field. The next section is "Business Name" with an input field. The final section is "Date of Birth *" with an input field.

Figure 3: Member registration of prakriti farming

build their urban farming business using the help of PrakritiFarming. But main disadvantages are lack of technology stack which can enable the members to operate automatically regarding the process flow during the growth of their gardens. Furthermore, The company has no technological advantages to allow its members to manage their garden via online. Instead of that, they always provide manual support of labor to the customers.

The blog page presented in the website also can't let users to post the articles via online. The admin of the web application can only update the blogs. Its great disadvantage of any registered users who wants post anything about rooftop gardening.

Blog



প্রকৃতির কাছে, 'প্রাকৃতি'র সাথে...

Test Test, February 22, 2017, আমার
গল্ল, 2

গাছ ও প্রকৃতি আর মানুষ, এরা সবাই
যেন ঠিক একই সুত্রে গাঁথা। একটু
ফিরে তাকালেই দেখতে পাওয়া যায়

আমার ব্যালকনি ও ছাদ বাগান

Test Test, February 20, 2017, আমার
গল্ল, 0

আপনি যেখানে অবস্থান করছেন তার
চারপাশে তাকিয়ে দেখুন –
ব্যালকনিতে, বাড়ির ছাদে, রেলিং,
রুমের মধ্যে, অফিস সাঁজাতে, বাড়ির
উঠানে...

Figure 4: Blog of Prakriti Farming

Ugao.com [13] is a complete e-commerce solution for urban farmers. It provides wide ranges of online services for the customers. It's website based on India. The website offers buying different gardening materials and services. Figure 2 presents the user interface of the application. In the Figure we can see the main features of the web application where user can see the buying option for the plants and accessories of the plantation, tools for nurture the garden, the pest-control information page and also the blog option for building an online community. The Buying option let users to order a

palnt or plant related product and lets user to pay the price via online payment methods.

Main core technology of the product lies on E-commerce capabilities.

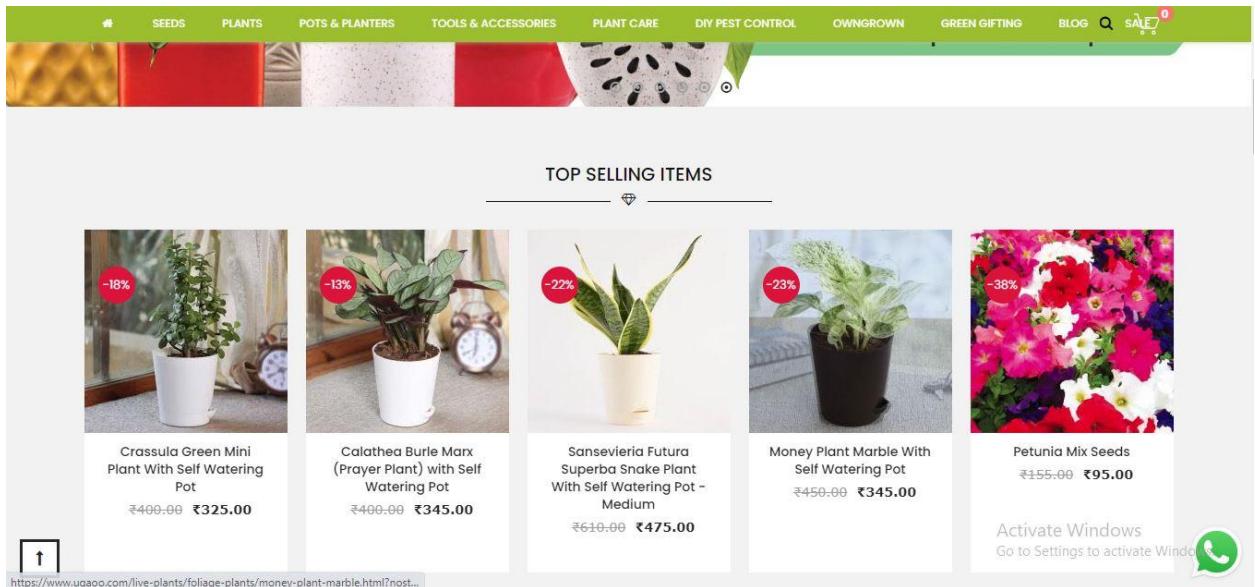


Figure 5: E-commerce Website of Ugao.com

The Ecommerce option for buying a product is shown in the figure 6. Here, a customer can order a plant via online and can pay via card or other online payment gateways.



Figure 6: Shopping Cart of the Ugao.com

The main technological disadvantages of the web application are the lack of features for the users who own nurseries and want to sell their plants via online methods. Interestingly, the main focus on urban gardening relies on the fast growth of online marketplace of plants of small business owners who can produce plants and can sell them via online. This business requirement is very crucial for addressing the implementation flow stack of the technology core in an urban plantation software.

SmatGardener [15] is an another product for developing a layout for a personal garden for the users who wants to grow their garden in their confined urban spaces. The software can be used via paid subscription and user can estimate the cost, plant growth and also the yield of the harvest. This options can be available via both of their online application and



Figure 7: Smart Gardener website

desktop based standalone application. In order to see the progress also monitoring charts which can be filled via manually by the users. The specialties of the software are the 3D gardening tool which can enable the users to the look of their proposed garden in the online platform. The figure 8 shows the overview of the product where we can see the motivation and objectives of the application.

The easiest way to plan, grow and harvest your own food.

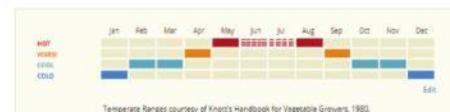
An online vegetable garden planner for anyone who wants homegrown, healthy and tasty food to be part of their lifestyle...while having a busy life.

[Subscribe Now](#) [Explore a Demo Garden](#)

ALL ABOUT GROWING FOOD, SIMPLIFIED

Personalized Vegetable Garden Planner

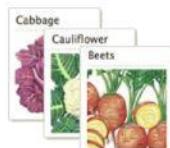
There are a lot of variables that go into planning a garden. Smart Gardener does all the hard work for you. We collect, calculate and create a smart personal profile of your garden just for you. Learn more about your profile.



[BROWSE](#)

[WHAT CAN I PLANT NOW?](#)

[+ ADD A NEW VARIETY](#)



The Right Plants for You

We make it easy to find the right plants, so you can't go wrong. With over 3000 organic, GMO free, edible varieties to choose from (and buy from our partners), Smart Gardener offers you recommendations along with super simple ways to find plants suited to your growing conditions. Learn more about plants.

An Optimized Vegetable Garden Plan

Get your garden plan so you can get out in the garden. Smart Gardener combines your selected plants, vegetable garden layout, and household size with complex planting variables, to help create a Smart Garden Plan just for you. Learn more about Garden Plan.



Figure 8: smart garden Blog

The main disadvantages of the application are the lack of connection between sensing devices with the gardens which can be used to monitor different environmental parameters of the rooftop garden.

SmartCultiva [14] is an application where user can monitor their garden using different IOT devices. They offer different types of nano-sensing IOT devices which are used to read different types of environmental factors regarding the garden. This option can let users to monitor the health of their harvest. Figure 11,12 presents the features of the application where they offers the temperature and moisture reading devices and the cloud based technology of the product.

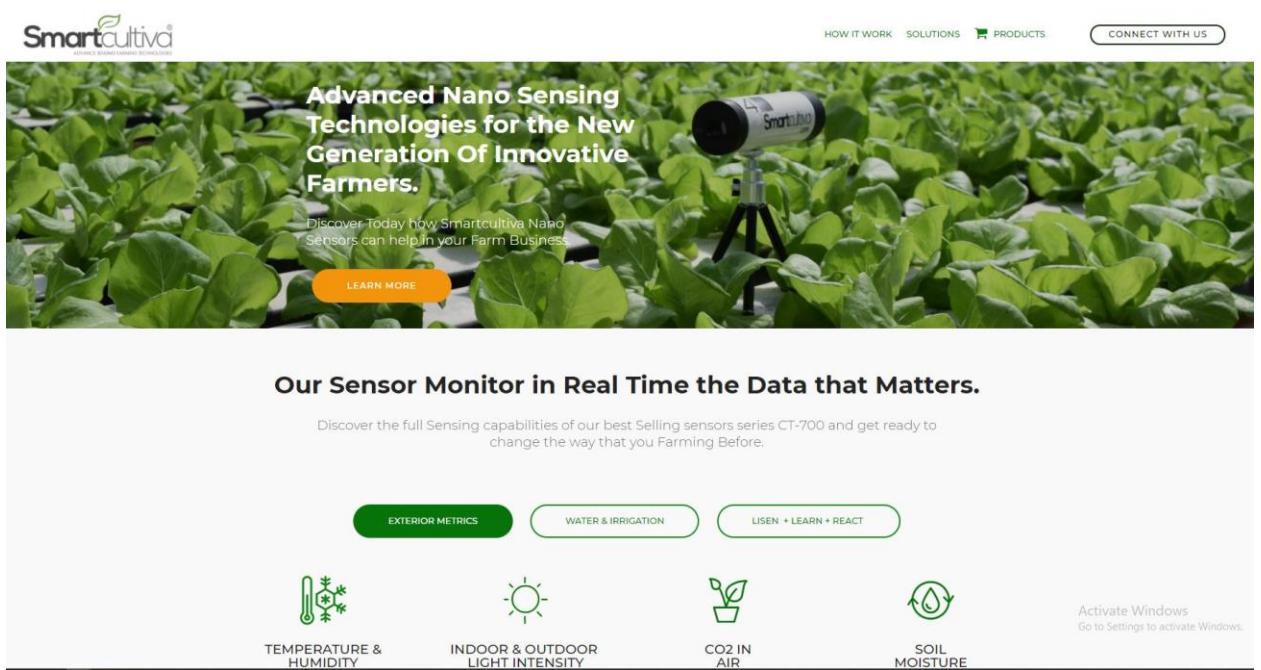


Figure 9: Smart cultiva website

The main disadvantages of the application are the lack of community support of the online garden lovers. The technology doesn't support blog posting, ecommerce solutions for the customers.

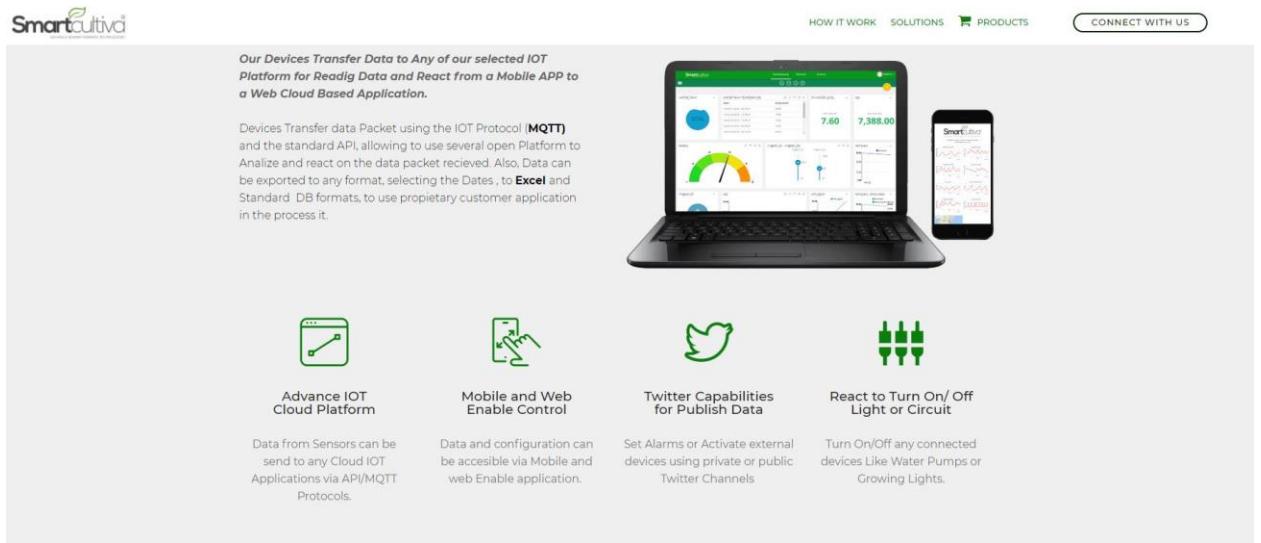


Figure 10: Smart cultiva website

In a nut shell, all of the existing solutions are very relevant of urban plantation. But, all of them have the motivation for helping urban gardeners. However, no such application is developed in the motivation for growing the online business for the nursery owners and the community buildup for the urban farming enthusiasts. In order to supplement this need of the community we want to propose a software project in form of both web-application and mobile application. This project primary aims for the users who wants to build their garden very easily and the users who wants to sell their plants as well manage their nursery inventory via online.

Existing Mobile Applications

There are many online applications is available on google Playstore. We have searched on Playstore and found a various types of online based rooftop gardening apps.

1. Rooftop Garden Ideas –In this apps, client will find only various types of design.as we know, Different people different interest .so people can find rooftop gardening design based on their taste, personality.

2. How to make a garden– This apps has the details of the basic knowledge of gardening .choosing the right soil ,how to make tub ,how to grow plant ,what's need to done for taking care of, scientific way etc.

3. Roof gardening Bangla- How to create a roof garden, basic knowledge of growing plants, tips about gardening.

4. chad krishi- Before starting gardening, how to prepare your roof or balcony, extra care, cactus planting, flower planting, different types of vegetables growing tips ,seasonal flower, fruits, vegetables growing tricks .

5. Rooftop Gardening: Pros and cons-at the first, basic need, how to plant a tree, how to take care, scientific way, easy way of gardening.

There are many more online based applications are available. We are mentioning few of them.

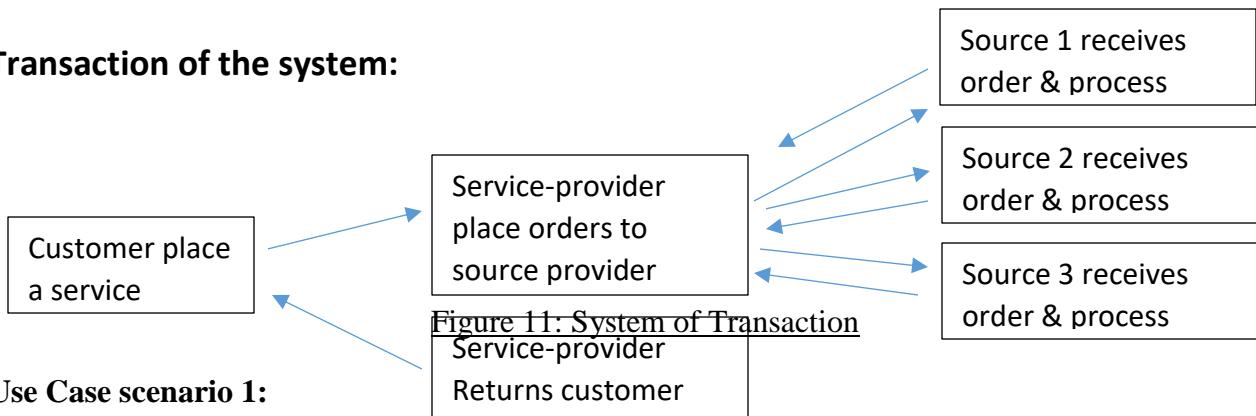
In our apps, we have all the solution. We are proposing many more solution for our client. They will be able choose products as per their interest through our app. Nowadays people are very busy with their life, they do not have much time for extra curriculum activities like gardening. This app will be blessing for them. They can place their order from us, more specifically from the nursery owner. We have feature of calling. Nursery owner information will be showing in the apps for the client. By seeing the advertising, they can choose the product. Customer don't even have to go the nursery; nursery owner

will reach the product to the client. These are the unique feature is available for our client. Also, gardeners can sell their vegetables.

2.2 Proposed Solution:

The urban gardening application is an application to provide gardening services to the customers who own the urban spaces. This application is a bridge between two communities: the plantation service providers who lack technological support and the city dwellers who want to grow garden in their urban spaces. At first we built our first prototype using the following use case scenarios:

Transaction of the system:



Use Case scenario 1:

User enters to the system as a guest user and then see the landing page of the online application. There user can browse the plantation tips forum and the information about the different service and source providers of garden materials like seed, soil, plantation bed and other equipment related to urban gardening.

Use case Scenario 2:

User wants to develop garden in his/her urban space. He/she orders a service provider's package. A package contains how a provider gives all necessary materials related to the incremental development of the garden. User then register as a Customer in the online platform in order to

place service order. He/she gets a customer id and phone number of the provider. Provider schedules an appointment to give the materials to the customer. Customer provides money to the service provider and starts to build the garden. Now the customer is a user of that service provider. They now can communicate with each other via the online platform.

Use Case scenario 3:

Service provider enters to the system as guest user to see all the necessary contact list of all the nursery owners/Person who sells gardening goods/person who sells soils/Person who sell pesticides (Sourcing provider). Now service provider can place order all the necessary materials to grow the garden for his customer.

Use Case scenario 4:

Service provider register into the system and place orders to different source providers. He is now a customer of the source provider.

In our first prototype, Service providers are the customers of the platform as they sell their services via the online platform. This online platform also offers freelancers to work with both house owners and plantation service providers so that house owner can get their plantation services from plantation service providers. The following diagram shows the main actors and agents with the list of general characteristics of the online platform:

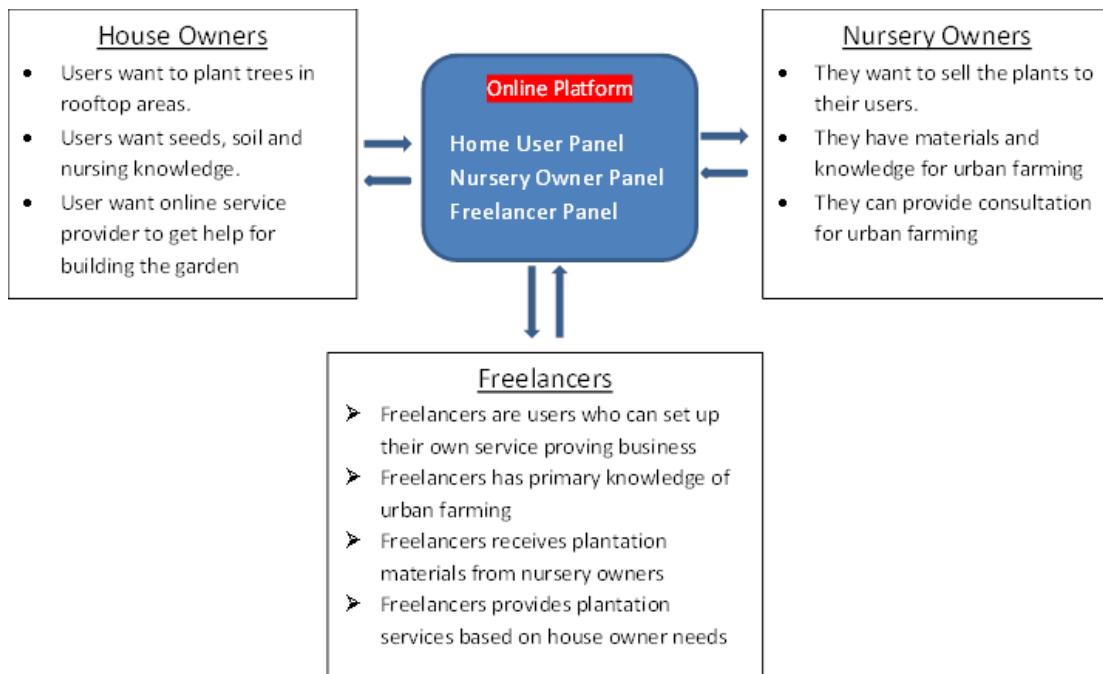


Figure 12: Actors and Agents of the Urban Farming Application

After developing the first prototype we decided to remove the service provider module and we build the provider module instead of service and source provider module. The use cases for the new prototype are:

1. User creates an order for a gardening service :

In this scenario, user wants to have some plantation program in his/her private spaces. In order to do this, user needs plantation material and plans to build his/her urban garden. For this user order a service from gardening service provider or independent nursery owner. The scenario in graphical view:

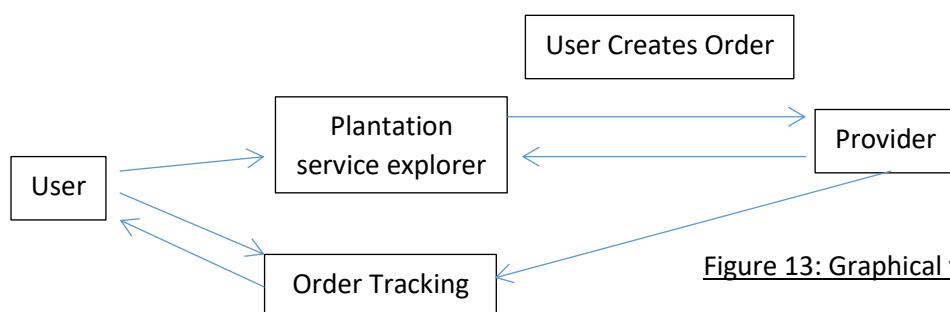


Figure 13: Graphical view of placing order

2. A Gardener creates a service model for delivering plants to their customers:

In this scenario, gardener makes a catalog of Nursery and then makes a service model for delivering service to the customer. User see the catalog of the gardener's nursery and the decide which service he/she needs to implement for his/her rooftop garden.

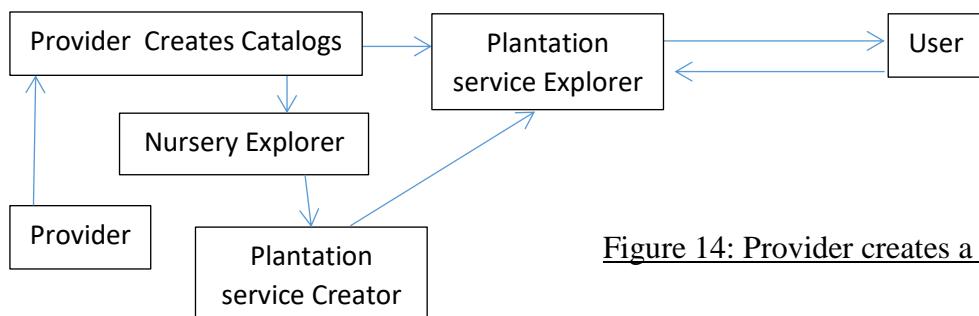


Figure 14: Provider creates a service module

Figure 15 presents the use case diagram of our final prototype application.

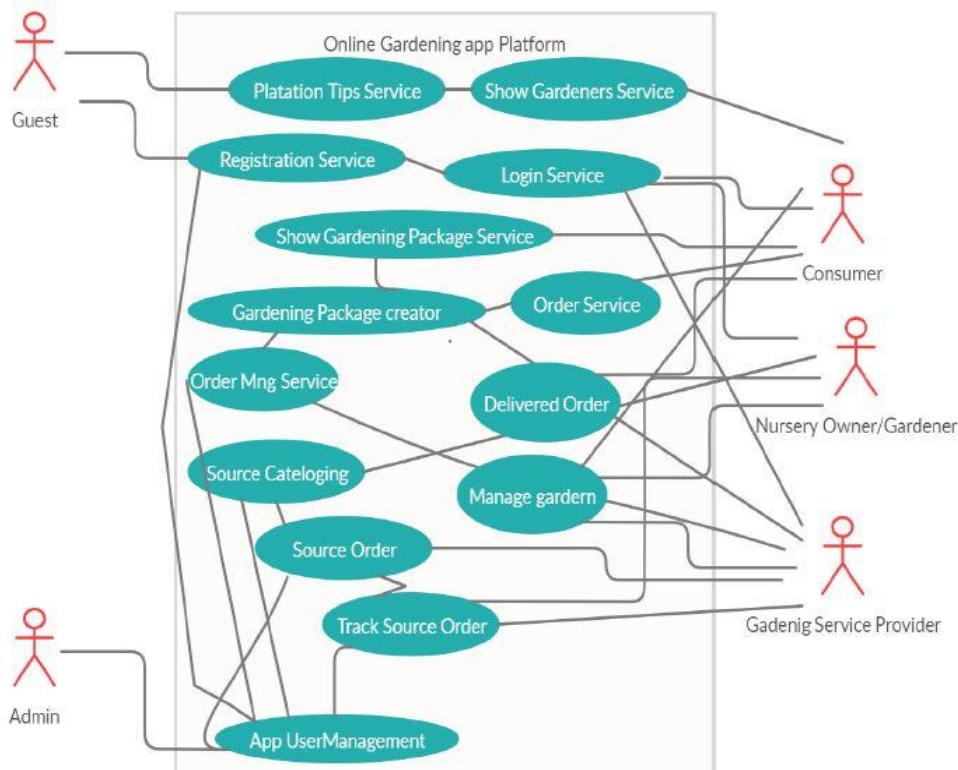


Figure 15: Use case diagram

Software design Patterns of the web application

The basic business logic is implemented using the service methods for each component. Each component may consist of single or several service. Service is a single class which can contain single or multiple method which has connection with single or multiple model class of the data modelling layer of the application layer.

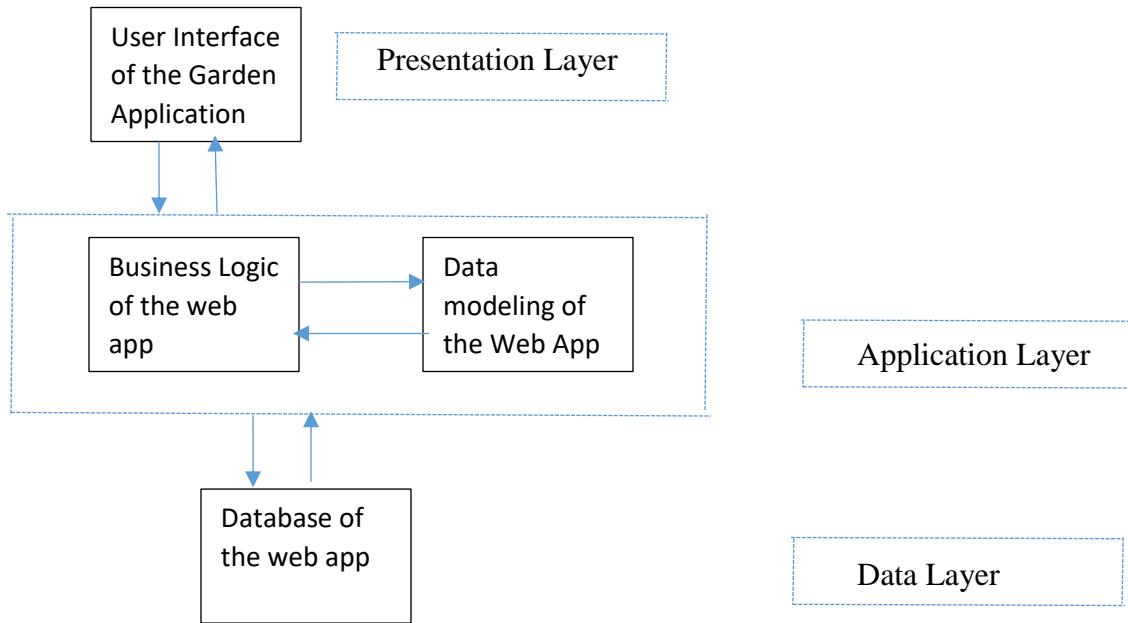


Figure 16: Top-Down View of the Application

A Single component of the web app maintains a specific feature of the web application which resides in the application layer. Single component can contain single or multiple services

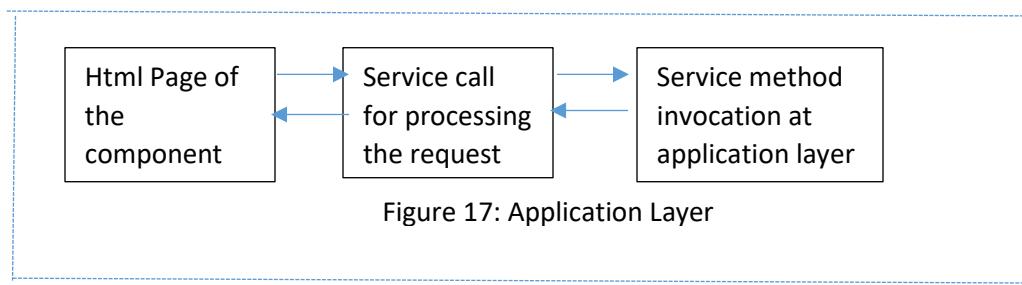


Figure 17: Application Layer

Technical details of the mobile application

Rooftops are the places of fantasy and imagination-places that sit above the din and chaos of the city, engaged with and yet apart from the city's motion.

Rooftop farming is usually done using green roof, hydroponics , aeroponics or air-dynaponics systems or container gardens. Besides using the already present space at the roof itself, additional platforms could possibly be created between high-rise buildings called "aero-bridges".

Thinking all these perspectives, this app goal is to create a linkage between all necessary people. These are option that will be found in our apps by our client .

- Rooftop gardening making(how to make or create a rooftop garden)
- Rooftop garden model
- Nursery owners
- Calling option
- Selling-buying
- Blogging

This apps will help people to find a design a garden that fits with their space needs, expectation. Gardens can take many forms. That's why our apps has an option called 'rooftop garden design and model. In this option, customer will find various types of design and model. They can find many forms from that. If they think this is not enough, they need more help then we will provide special services like expert people, nursery owners etc.

Then we have, 'Rooftop garden Making ', in this option, customer is going to get knowledge about how to create a rooftop garden. As we know, in urban areas, there's limited space that is necessary for building a garden. If a customer think that they are running out of space, or if they want an outdoor living space, or in a commercial environment or in an office, then this option is for them. Rooftop gardens are an ideal way for an urban gardener to increase their space. Rooftop gardens also make good use of unused place. For this, a few things to keep in mind if a customer wants to build up or create a rooftop gardening.

First of all, we need to know what the special requirements of a homeowner are. Some people wants to start with mainly flowers and some fruits trees. Some people wants to start with planting vegetable to have fresh and nutritious foods close to their hands. So following customers demand, experts will suggest them. Not only that, as space is limited, experts have to keep that in mind when designing a new garden. What are size or which type of garden that a customer needs or wants. If a customer is a beginner, the basic parts of making a garden will be covered in our apps. He will get enough knowledge as a beginner.

Second thing to keep in mind that get a contractor involved. We will be the 3rd party to find a contractor. Contractor will tell them if the building is safe to build a rooftop garden on. Some buildings is not designed to withstand the additional weight a rooftop garden would add. Maybe some buildings are able to take the extra weight. A contractor should be able to tell them if this is the case with their building.

Another option is 'nursery owners information '. In this option, customer will find nurseries trees, flowers, fruits etc. Nursery owners can advertise their own items so that people can decide

from where they will purchase. We have talked to a few nursery owners. They agreed to join with us .so those who agreed their product details will be available in our apps.

Not only their nursery details but also customer can call the nursery owners. Whomever they decide to purchase from, customer can call from their phone .they can have their queries to nursery owners like what trees are available or how they will delivery the products. Customer has to pay the price.

Another important service that will provide is ‘selling and buying’. Anyone can be a business. In this option, a simple housewife also can start or earn money. In their garden, someone’s growing a vegetable in their gardens, they want to sell them. In this opinion, they will be able to do that.

After starting the garden, customer will find various problems like insects are attacking their trees, plants, fruits and flowers. Some plants are dying as insects are the most common cause of garden damage.

Spring is a fantastic time in the food garden. Tiny seedlings poke through the soil. Newly set out transplants raise their leaves to the sun. Everything is interesting- when you find out that holes chewed in leaves or plants wilting and yellowing. A pest or insects or disease is at work! Now what to do?

It’s safe to say that sooner or later every gardener will find a pest or a disease problem affecting plants in their garden. Fortunately, there are some easy strategies you can follow to make pest and disease problems less likely to cause and less damaging when they do. Insects are certainly major pests of vegetable crops. But other groups of animals can wreak havoc as well.

For this type of problem, “insects and pests control “section is up for this. Customer will able to find out the problem. Familiarizing yourself with the pest and disease problems that you are likely to encounter is a first way to coming up with a pest and disease control plan.

Another addition is “Blogging”. People who are green lovers can share their experiences, love for plants, new plants or trees through Blogging. Through Blogging, it will create a community between green lovers and nursery owners.

We also keep the security part in our mind, those who will registered, they have to provide the national I’d card number, full name, address etc.

Feature description of mobile app:

The world today appears to run on apps. If it is something that individuals can associate with, then it’s likely that people look for an apps that is easy to use. Well planned apps are attractive and make you want to utilize them, while ineffectively planned one are simply irritating and don’t fulfil the requirements of people. While a few essentials, for example, ease of use, usefulness and format help the client experience, one of the most griped about issues with applications are the highlights, be it absence of expected ones or terrible utilization of those which are there.

When you have your essential thought, the design, and shading plan right, you have to begin contemplating what the client really needs, since, supposing that you don’t address their issues, a rival may, and thus you could lose huge business. There are a couple of highlights which are ideal to have and some which are unquestionable requirements, yet many will make your application truly stick out. By keeping all these fact in our apps so that people don’t feel bored

using our apps. People don't like complicated thing especially some things what they will use in their regular lives. Apps should be very easy to use, need to fulfil some basic needs of clients.

So, we have created some feature for our apps for our clients.

Modesty

Numerous individuals have limited capacity to focus. So if you make your application hard to explore or complicated and it's very tough to understand at that point they will lose their interest. If your client can't get to their data rapidly and effectively as per their need or interest, they will get baffled and do it another way – potentially by utilizing another application. Clear, cleaned up screens with evident prompts the following stage in the application cycle and no uncertainty improve the client experience and will urge them to keep on utilizing it. In our apps, it will be easy to use for clients. They won't be bored while using it. While entering in the apps, users have to submit the information about himself/herself, We keep this part for the security purpose. Whatever information they need about the gardening , they will be able to get through it .

Speed

Quick stacking screens are essential. Nobody likes pausing, especially when all they need something to know or looking for something and they see this at is a screen loading and this is very disappointing and irritating for a user. If they need to wait this long, they will look for something else which is better than this. Speed implies a proper arrangement of illustrations and not bringing enormous tables and information bases. Keep it basic and keep it fast.

picture. We tried to maintain a great picture where client won't find any obligation to see the pictures.

2.3 Solution Assessment:

After the development of our prototype we compared our project with the existing solutions we discussed in the earlier sections. We present table 2 to demonstrate the comparison between our system and the existing solutions.

Website/Technology	Main Features of the existing system	Available Features in UrbanFamingApp compared to the product
<u>NagarKrishi</u>	<ul style="list-style-type: none">▪ Shopping of garden kits.▪ Online Community	<ul style="list-style-type: none">▪ Yes, Here also this feature is available.▪ Here, In our app we can also build online blogs.▪ Extra feature: Selling platform for the nursery owners.
<u>PrakrityFarming</u>	<ul style="list-style-type: none">▪ Membership based service for garden	<ul style="list-style-type: none">▪ We also care for membership based garden management via

	management	online
<u>Ugao</u>	<ul style="list-style-type: none"> ▪ Garden services, ▪ Online shopping of garden kits ▪ Online Payment is possible 	<ul style="list-style-type: none"> ▪ All the services that Ugao offers, right now all of them are not available. ▪ Payment options also not available
<u>SmartGardener</u>	<ul style="list-style-type: none"> ▪ 3D garden designing 	<ul style="list-style-type: none"> ▪ 3D design of garden space is not possible in the application
<u>SmartCultivaApp</u>	<ul style="list-style-type: none"> ▪ Sale of IOT devices for urban farming ▪ Smart sensors connections of urban farming 	<ul style="list-style-type: none"> ▪ Smart sensors of urban farming is not connected in the application

Table 2: Table of comparisons between existing solution and our solution

2.4 Design Alternative:

After developing the first prototype we conducted a survey. After the experimental data analysis, we concluded that we need a redesign in our software product. Then we developed a new prototype for the urban farming application. The table 3 is presented about the difference between the new and old prototype.

Main Features of the Prototype Version 1	Main Feature of the Final Prototype Version
User can add plants via user module	<u>Design Alternative:</u> User now only can add new plant information if he/she own a nursery business
Service provider module only offers service to customer for the garden management	<u>Design Alternative:</u> After the survey and physical visit to the nursery we concluded that we don't need individual workers who want to give gardening services to the customers. It's just economically feasible. So we removed the module in the final prototype

	design.
Source Provider Module enable the service providers to process an order of a customer	<p>Design Alternative:</p> <p>Source provider is not needed since we removed the service provider module. So we newly introduced Provider module which can be operated by the nursery owners</p>

Table 3: Table of comparisons between First prototype and our Final Prototype

2.5 Technical Design: Module Level

In the first prototype implementation, we developed our project via dividing the project into 5 major modules for the web application. These modules are presented in Table 2:

Module Name	List of the components
Basic Module	LandingUIService LoginService SignUpService FaqService SearchService RatingService
Consumer Module	ConsumerLandingUI ConsumerBlog ConsumerFarmService ConsumerOrder GardenerTracking ConsumerIncomeManagement
Service Provider module	ServiceProviderLandingUI PackageManagementService SourceManagementService UserMangementService PlantationBlog ActivityService ProviderIncomeManagement
Source Provider Module	SourceProviderLandingUI CatelogManagement SourceProviderIncomeManagement
Admin Module	AdminUI UserRoleManagement UserRatingsValidator UserPrevilageManagement

Table 4: List of modules of the First Prototype

The interaction of different webpages of the web application are presented in the figure 13.

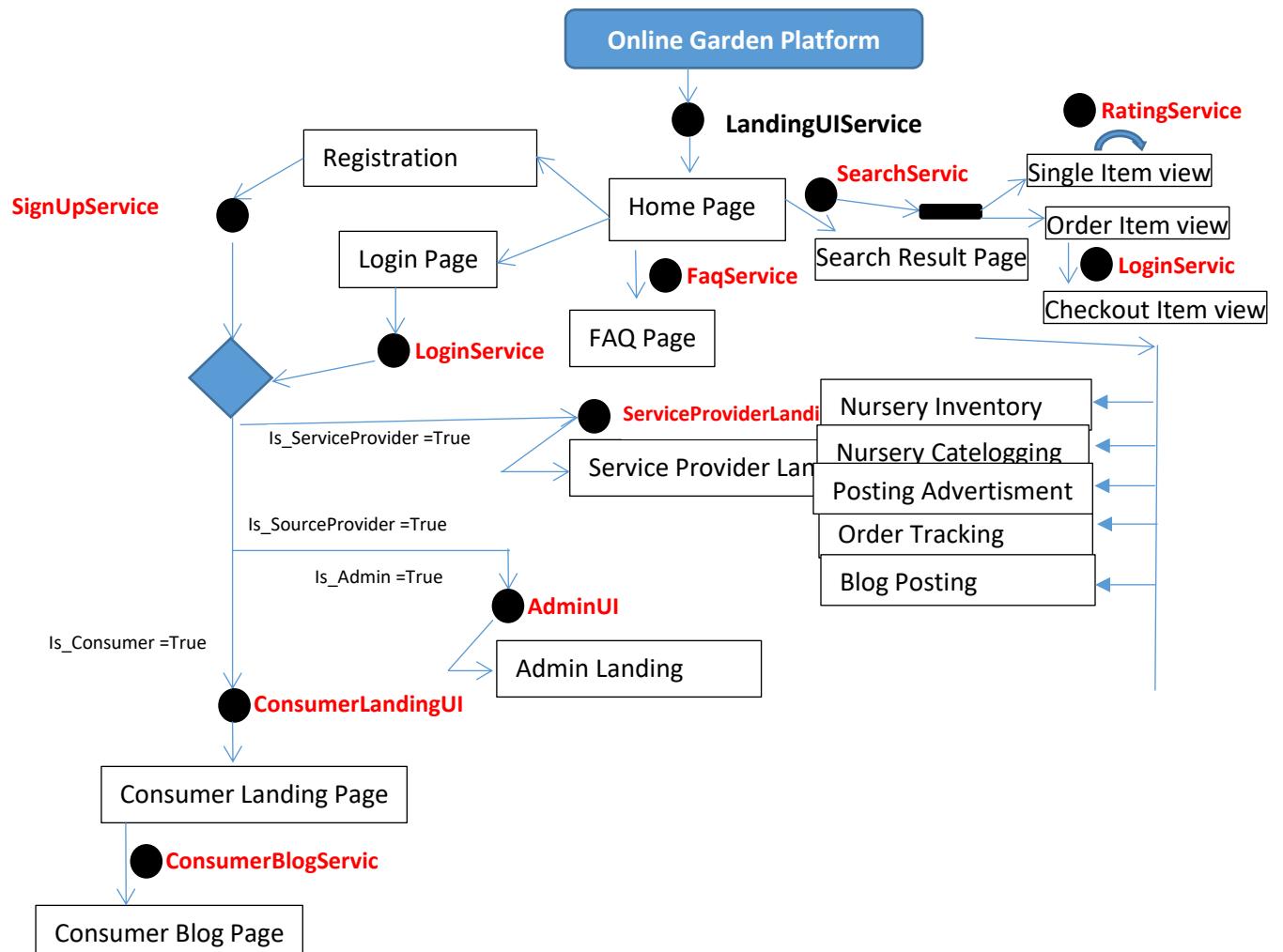


Figure 18: Activity Diagram

After the development of the first prototype we decided to remove the service provider and source provider module to replace these with new provider module. Our newly developed final prototype has four basic modules:

For the web application:

1. Admin Module:

This is the module where server/app administrator manages the user roles, privileges and other administrative tasks of the web application

2. User Module:

This is the module which is used by 3 types of user category:

Customer, Nursery Owner/Gardener, Service Provider

3. Authentication Module:

This module helps the users or administrator to validate the form data or other data driven functions.

4. Data Management module:

This module helps the end users to manage their information also this module can predict/suggest new information via its data visualization capability.

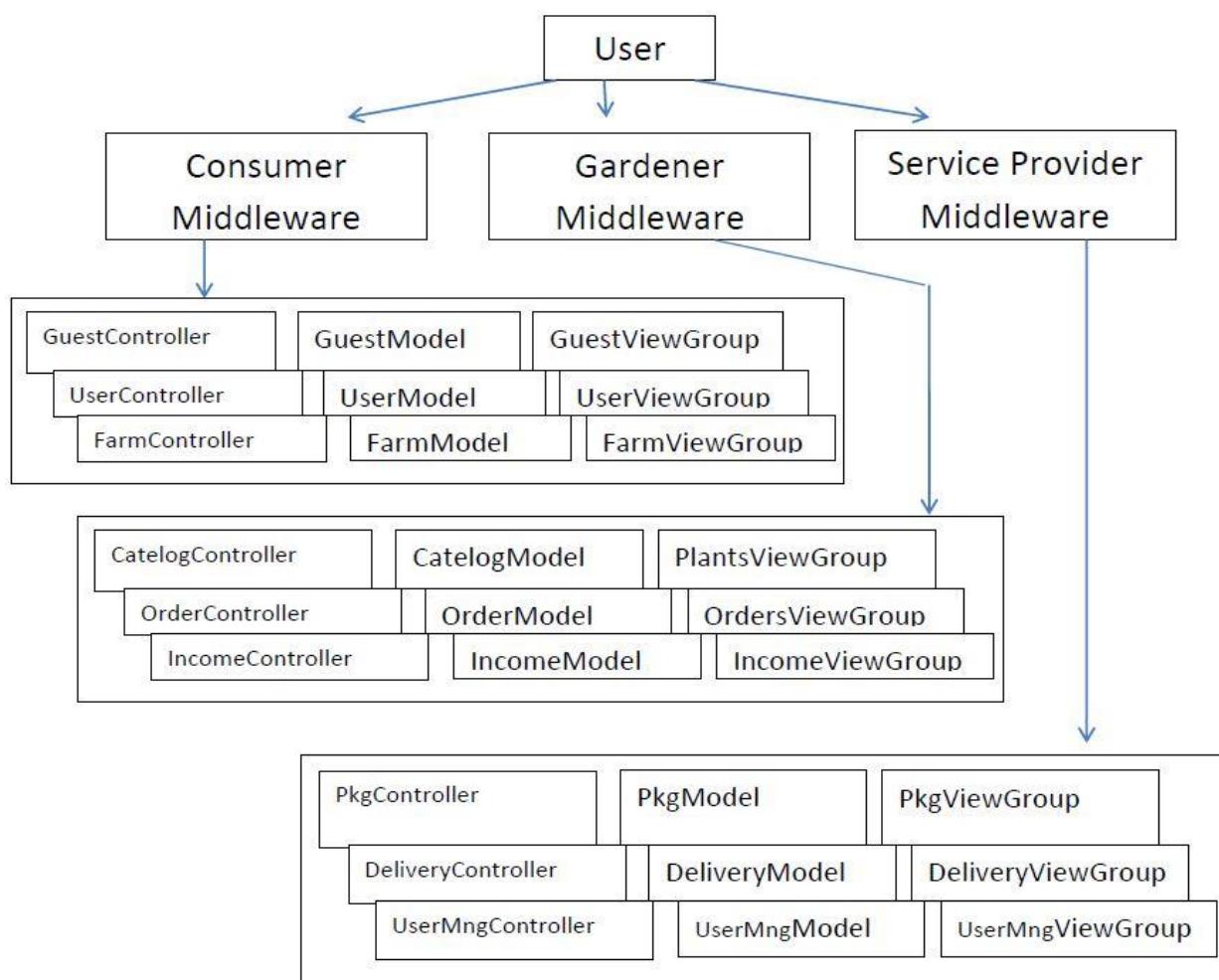


Figure 19: Block diagram of user module

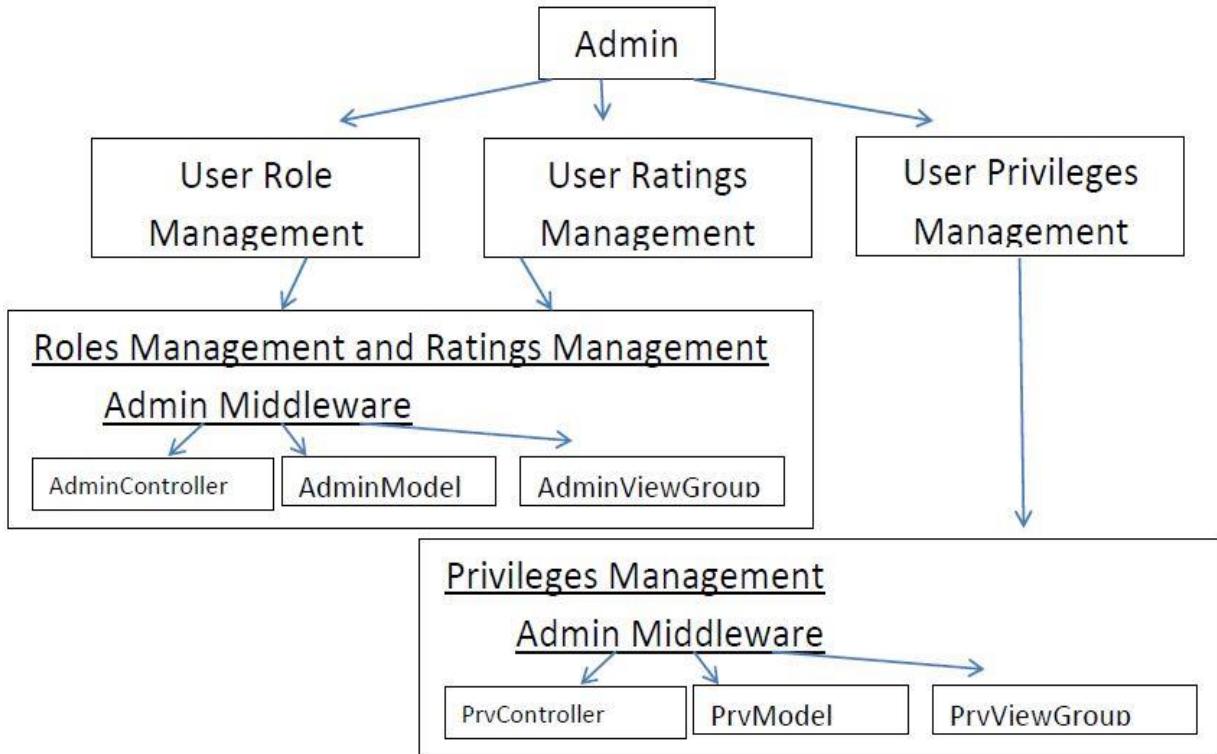


Figure 20: Block diagram of admin module

Details of the Service Provider Module:

List of implemented features:

1. Service Creator: Using this feature a provider creates list of packages were each package has list of services. Provider can also set the package cost using a form.
2. Source order: Using this feature a provider place a order in the system to source a plantation material.
3. User Management: Using this feature Provider see the list of customers who are invoking different packages. Provider can manage activities of each customers garden using this tool via customer id (In that case a customer id is assigned to the specific provider id)
4. User Preference List: List of customers based on the preference. It helps the service provider to create their service package.

5. Plantation Blog: Using this feature service provider can create article about plantation.
Using this feature provider can also give free consultations.
6. Service Management: Using this feature user can update the contents of the packages that are published in the site.
7. Payment page: Service provider has to pay the sourcing provider and using this feature provider can pay via payment gateways.
8. Activity monitor: Using this feature provider can track an individual sourcing order.
9. Accept Sourcing: Using this feature provider can accept the order from the customer and then collect the source from the sourcing provider and then deliver it to the customer. In this way, provider gets a delivery charge.
10. Income management: Using this feature provider can connect his online account to bank account/Bikas account. There will be 2FA authentication for using this feature. Using this feature provider also can make payment from his income.
11. Profile Management: Provider can upload his information like National Id and permanent address. In this feature provider can also update his info. For security purpose he needs to be verified himself.

2.3 Technical Design: System Level

Our web application for the urban farming has modular design principles in the first prototype development of our application. The system level design of the first prototype is presented in the figure 19.

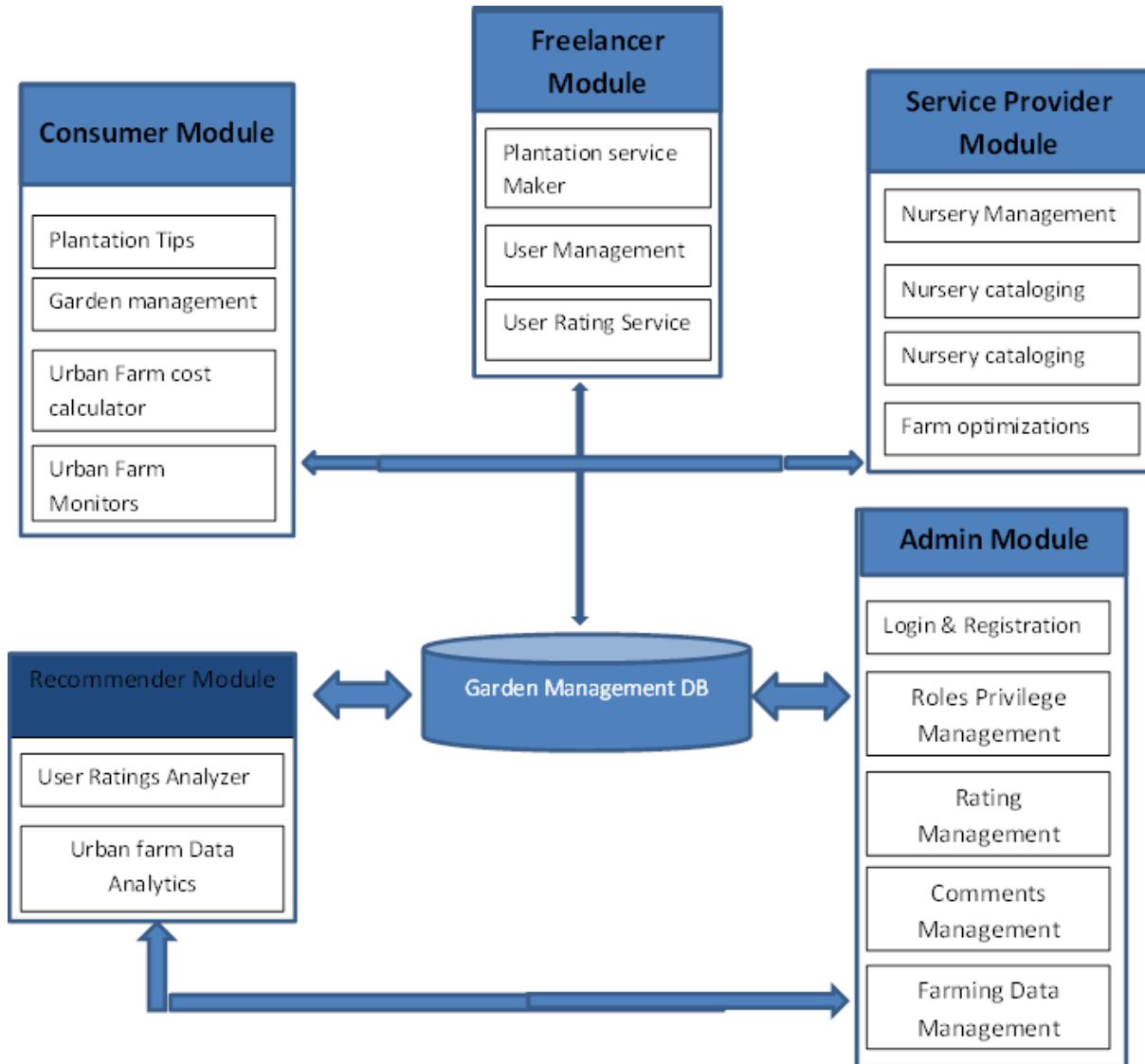
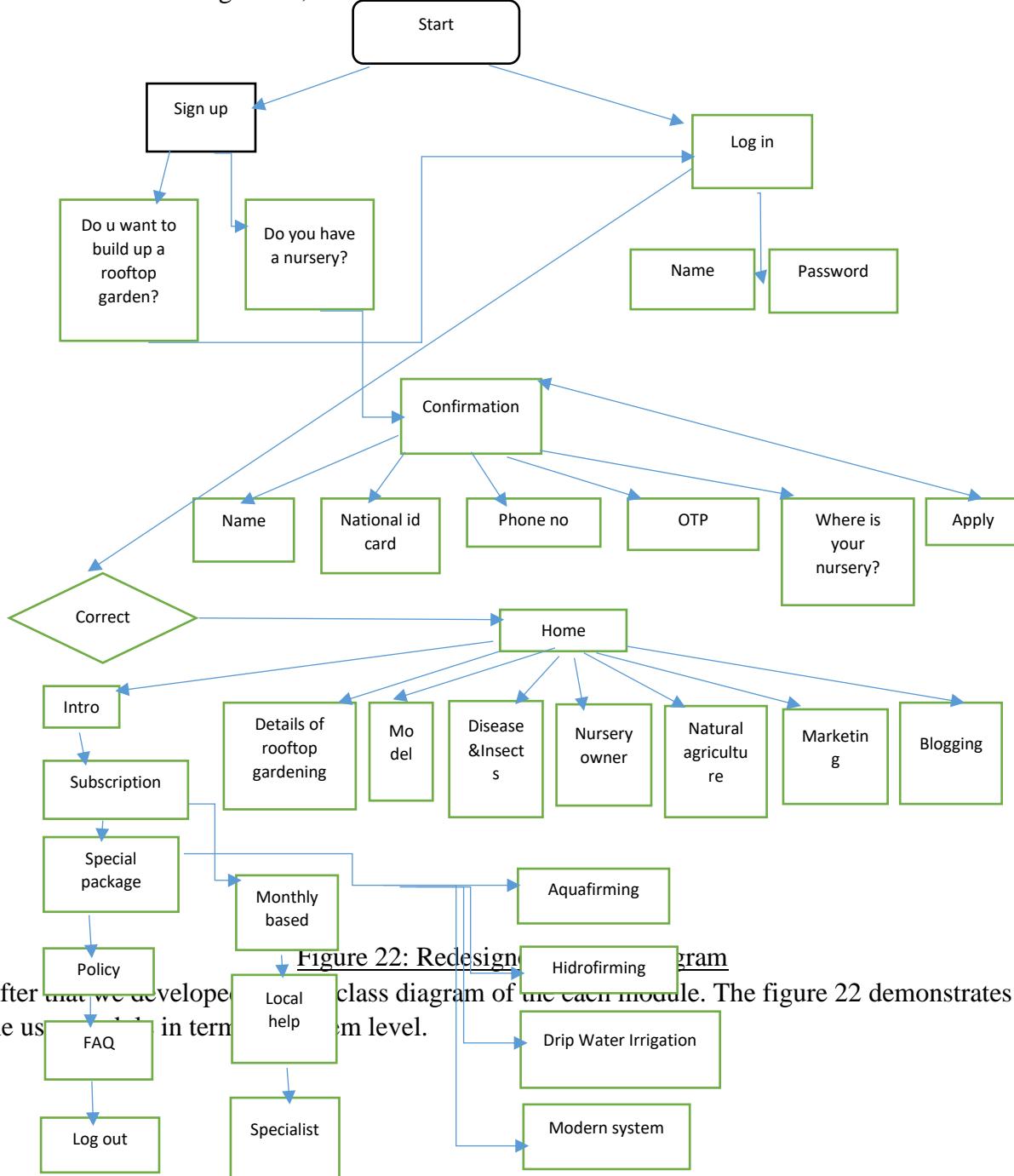


Figure 21: System Level Design of the Software

After the development of the technology stack of the first prototype, we run several UI tests and found that freelancer, service provider module and recommender modular is not necessary at this time.

time. So we redesign our product to accommodate our new process flow of the application demonstrated in the figure 17,



the us

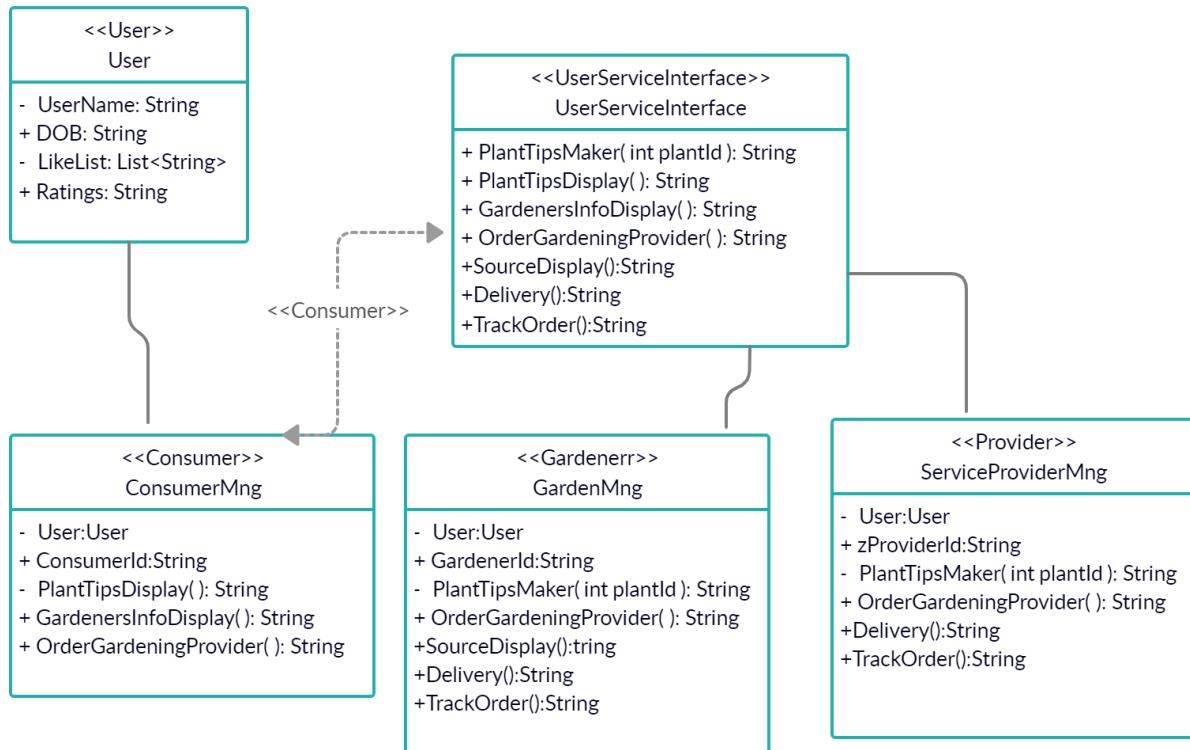


Figure 23 :UML Class Diagram of the User Module of the Web Application

The project have database schema which is presented in the figure 25. Application architecture of the mobile application is presented in figure 24.

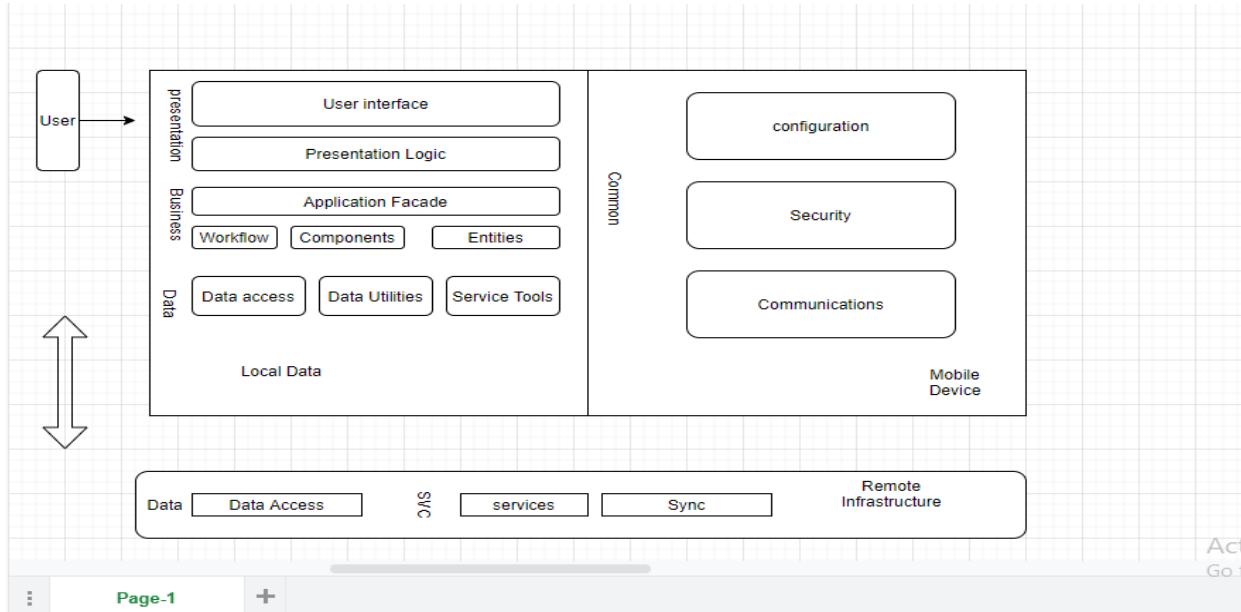
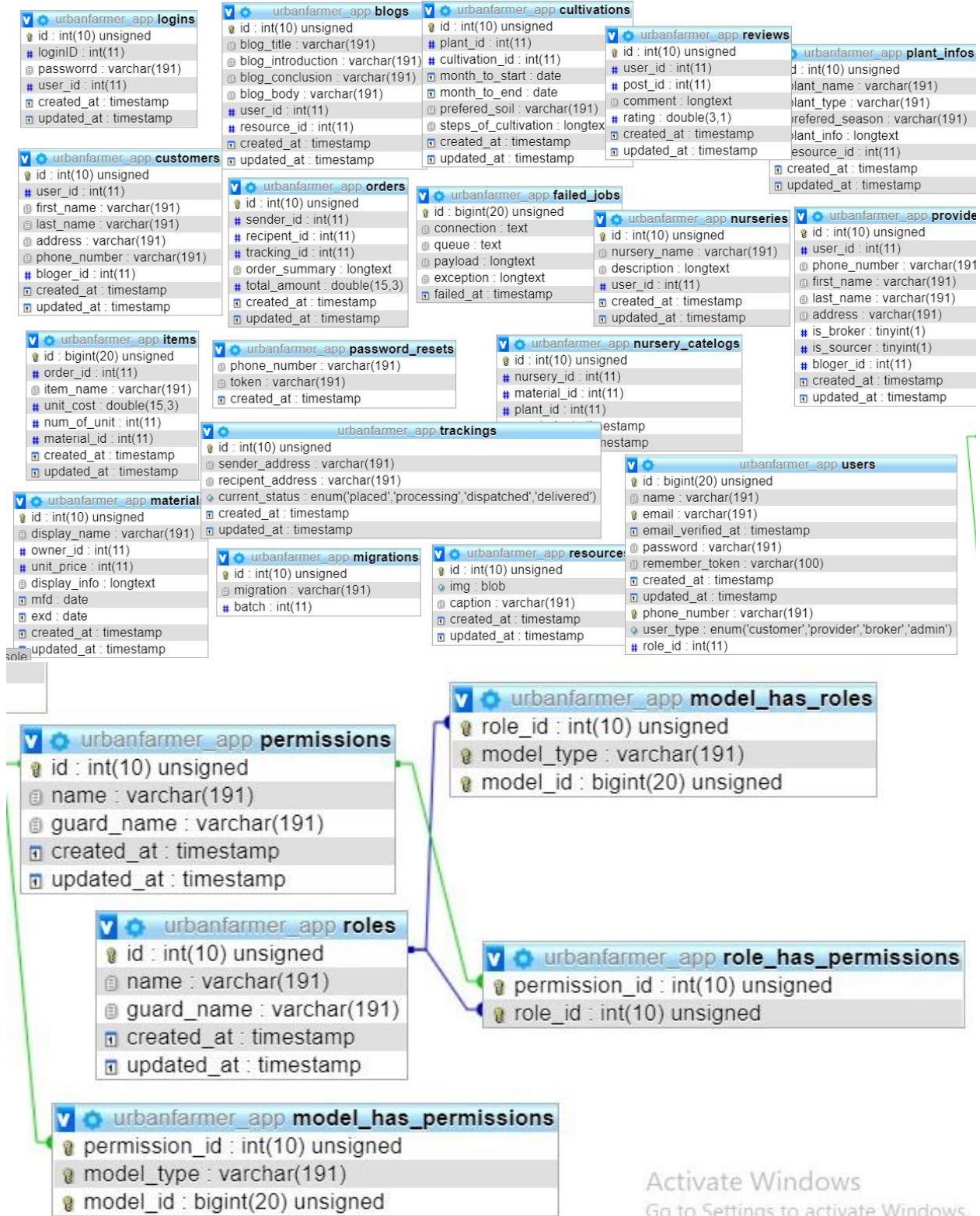


Figure 24: Mobile Application Architecture



Activate Windows
Go to Settings to activate Windows.

Figure 25: Database Schematics

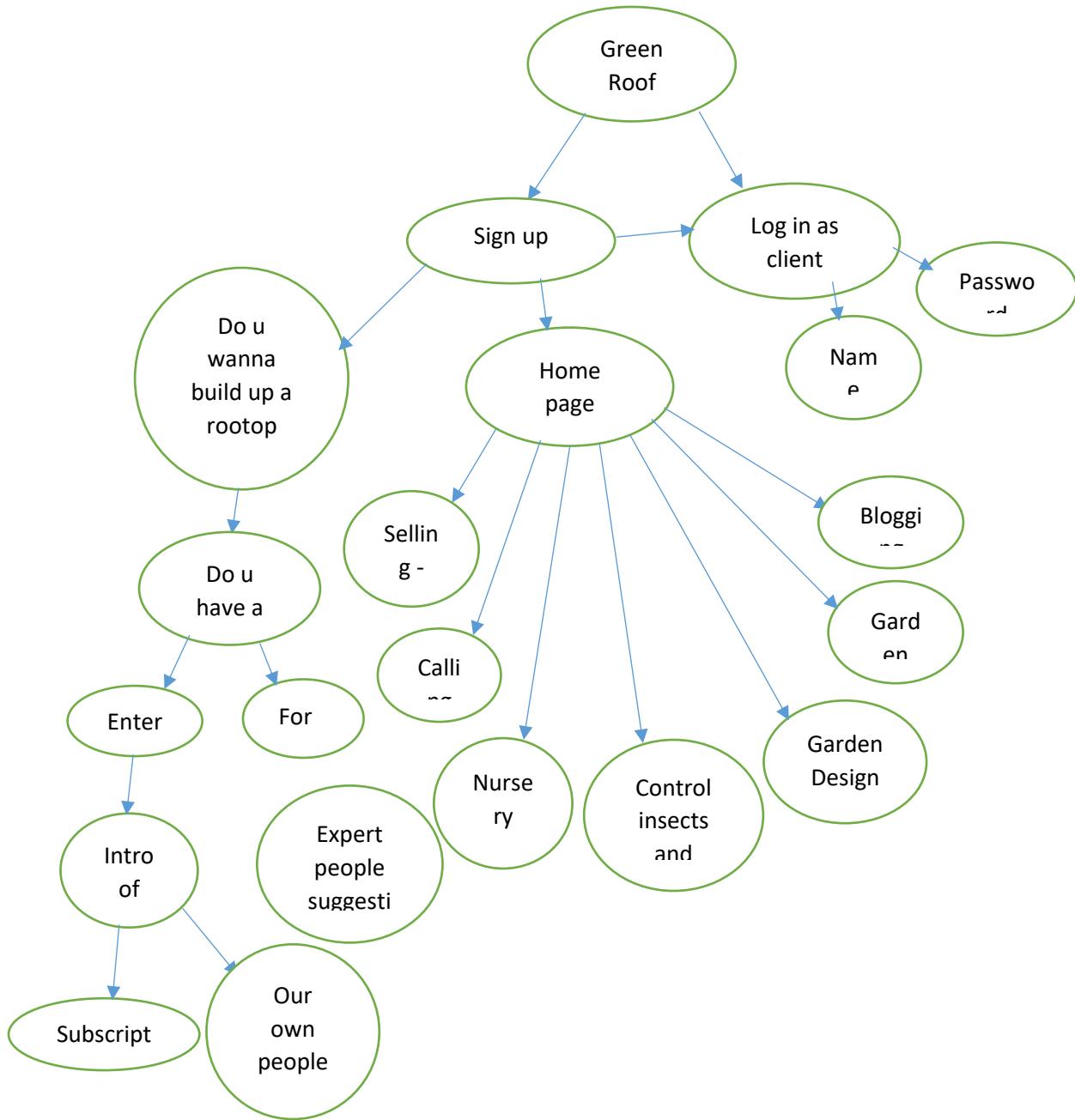


Figure 26: Application components/ Activity diagram of the mobile application

In the above Figure 26. The mobile app activity diagram is shown where we can see the code flow of the application.

CHAPTER 3

ESSENTIAL PARTS AND DEVICES

3.1 Description of Components:

Required skills and Selection of Software for the Development of Online rooftop gardening application

Online Urban Gardening is a software solution to the customers where awareness about rooftop gardening, ordering gardening materials for the interested city dwellers, making a plantation business etc. made possible by the developers. The major required skills for the development of the inline gardening application platform are to understanding the popular software design patterns, database design, User interface development, mobile application development, Learning module development.

The following demonstration explains the above required skills:

Database Development:

To implement the proposed solution concepts about the RDMS databases should be revisited by the developer. In the case we implemented MYSQL DB, FIREBASE DB and Laravel Eloquent ORM for implementing the database of the online gardening application as well as the object relational mapping for the communication between the DB and the software itself.

MYSQL DB:

We used MYSQL 5.0 for the DB development and apache web server to maintain the mysql database. For the DB development the RAW SQL queries for DB Table creation are needed to implement at first stage of the application development.

FIREBASE DB:

It's a very good database for the mobile app development of the online gardening application. The firebase DB of the Platform is integrated with the android application which is also connected parallel to our web app solution.

React Native

React Native is a structure that empowers web engineers to make strong portable applications utilizing their current JavaScript information. It offers quicker versatile development and more proficient code sharing across iOS, Android, and the Web, without relinquishing the end client's insight or application quality.

Node.js

Node.js is an open source stage based on Chrome's JavaScript runtime; it offers an approach to handily construct quick, adaptable projects. Node.js permits you to run JavaScript in Terminal, and makes modules.

Laravel Eloquen ORM:

Object relational mapping is a very important required skill to maintain a scalable database. In the PHP Laravel application of the online gardening platform can be integrated with the MYSQL DB using the ORM language Eloquent. Using this language we can create new relational data programmatically and import those data objects into relational data for the MYSQL.

Popular Software Design Patterns:

During the development phase of the Online urban Gardening Platform we learned about the Model-View-Controller concept of software engineering. Model is the object related entities which are responsible to represent data. Every table of the

plantation database is considered as entity of the system. Each entity has several operations to be made to perform CRUD (Create/Read/Update/Delete) operation and the entities are can be defined/mapped using the Model class of the application. Model class also has the operational ability to perform custom data access which can be related with one or more table of the DB. Model class can be accessible via Controller class. Controller class is the core feature of the middleware concept. We need to implement the middleware concept into the application development as middleware contains one or multiple classes which are responsible for handling multiple operations. View is the concept where User Interface can be visualized. Basically Views are the HTML /PHP pages which are responsible to output results of user oriented tasks which are described in the feature list of the application.

User Interface Development:

UI development is very important to implement a good graphical design for the customers. To make the UI more user friendly the Bootstrap framework is required. Bootstrap is a CSS framework which is a very powerful tool to implement robust UI design. In our application development we also required to learn the Blade engine of Laravel application framework. Blade is the template engine of Laravel framework. We also require to learn XML for the development of the UI of the mobile application.

Mobile Application Development:

Android OS is a very popular OS among all the smartphone users around the world. So implementation of the mobile application in the online gardening application is really required. To implement the solution we require Android studio for the

Application development and Rest API for the interconnectivity between the mobile app and the web app.

Overall, The sites that are helping us to learn these technologies:

1. <https://www.javatpoint.com/php-mvc-architecture> (22.11.2019)
2. <https://laravel.com/docs/6.x> (23.11.2019)
3. <https://www.w3schools.com/php/> (24.11.2019)
4. <https://www.codecademy.com/learn/introduction-to-javascript>(25.11.2019)
5. <https://jqueryui.com/accordion/>(25.11.2019)
6. <https://websitesetup.org/bootstrap-tutorial-for-beginners/> (26.11.2019)
7. <https://dev.mysql.com/doc/mysql-tutorial-excerpt/8.0/en/>(26.11.2019)

3.2 Test Requirements:

Hardware tools are basically needed in software solution when IoT devices are introduced. In our problem domain IoT devices are insignificant in terms of offering services to its customers. Primarily, our software solution is for ordering and maintaining the garden in rooftop spaces in urban areas. For implement this problem domain the system doesn't require any IoT devices. Although we need to implement a smart system such a way that a gardener can get maximum amount of gardening output. If we implement our solution in this extend Humidity sensors and other sensors related to agriculture is required and the real data should be visualized as smart features of the system.

CHAPTER 4

WORKING SHEETS

4.1 Work Breakdown Structure:

During the development of the project we followed software development processes. At first we modeled our phases of the application development. Then we developed each stage via incremental manner. The list of phases for the urban gardening applications below:

List of Phase:

- 1) Requirement Engineering
- 2) Project Structure Setup
- 3) Development stage
 - a. Selection of necessary HW/SW requirements
 - b. Selection of coding environments for the project
 - c. User interface Design
 - d. Business logic Implementation
 - e. Database Development
- 4) Deployment stage
 - a. Integration stage
 - b. Prepare the project for local application server
 - c. Deployment the solution into live network
 - d. Testing the project with demo users
 - e. Testing the project with real users
 - f. Analysis of the test data

Details of the phases:

Phase 1: Requirement Engineering

In this stage, we prepare the list of requirement for developing a potential software product for urban farming. In this stage, we prepared a survey to understand the need of urban farming and the modes of treatments needed to build a scalable online urban farming platform.

Phase 1 tasks:

1. **Draft survey design:** We made a survey questioner to gather knowledge about the requirement of the potential customer of the platform. Our survey has been done via two steps:
 - a. Questioner setup

We designed the following questioner:

Experimental Data Analysis

Gender:
<input type="radio"/> Male
<input type="radio"/> Female

To know about the scenarios regarding the urban farming and more user friendly and optimized system development, I propose a survey for getting outline information from

Occupation:

Age:
<input type="radio"/> 10 – 20
<input type="radio"/> 21 – 30
<input type="radio"/> 31 – 40
<input type="radio"/> 41 – 50
<input type="radio"/> 51 – 60

the user end:

1. What is your **primary reason** for rooftop gardening/farming? (**Choose only one**)

- Caring for environment
- Mental peace & happiness
- Hobby
- Passing time
- Source of fresh food (fruits & vegetables)
- Economic reason (to earn money)

- Others _____
2. What are your **reasons** for rooftop gardening/farming? (**You can choose multiple options.**)
- Caring for environment
 Mental peace & happiness
 Hobby
 Passing time
 Source of fresh food (fruits & vegetables)
 Economic reason (to earn money)
 Others _____
3. What types of plants you want to grow for garden? (**You can choose multiple answers.,**)
- Decorative plants
 Expensive plants
 Flower plants
 Fruit plants
 Vegetable plants
 Medicinal plants
 Most oxygen producing/ CO2 absorption plants
 Crops (for farming)
 Grass
 Others _____
4. What is the **primary plant type** you want to choose for your garden? (**You can choose only one option).**
- Decorative plants
 Expensive plants
 Flower plants
 Fruit plants
 Vegetable plants
 Medicinal plants
 Most oxygen producing/ CO2 absorption plants
 Crops (for farming)
 Grass (for farming)
 Others _____

5. If you are offered a service where service provider will **setup the garden (provide seeds, soil, fertilizer, initial maintenance)** for you, how much would you like to pay for that service?

- Less than 5000 taka
- 5000 taka – 10000 taka
- 10000 taka – 20000 taka
- 20000 taka – 30,000 taka
- 30000 taka – 40000 taka
- 40000 taka - 50000 taka
- More than 50000 taka
- Others _____

6. If you are offered a service where service provider will **do the daily maintenance (watering plants, spreading fertilizers and pesticides, Overall maintenance) for your garden**, how much would you like to pay **monthly** for that service?

- Less than 1000 taka
- 1000 taka – 2000 taka
- 2001 taka – 5000 taka
- 5001 taka – 8000 taka
- 8001 taka – 10000 taka
- More than 10000 taka
- Others _____

..... **End**

This survey will help us to understand the facts of urban farming and developing suitable features of the current proposed system.

b. Physical inspection of the nurseries.

2. Final Discussions of the survey

3. Survey data analysis.

Phase 1 Mobile app survey Features

Statistical surveying: No business begins suddenly. Organizations regularly use overviews to see whether there's interest for an item or administration. An overview is likewise an apparatus to check what individuals from an objective market consider new thoughts. So, we put a survey on numerous number of people that what they think of online gardening. Not only common people whom are garden lovers but also the nursery owner. We tried to make them understand about apps goal. We find very positive review from them.

Get item input: With survey, organizations can discover what clients think about their items. Surveys are an incredible method to comprehend which parts or method clients use the most and about their expectations. They likewise assist organizations with discovering what updates are important. Through our survey it really helped us to improve our services.

Track consumer satisfaction: This survey helps us to connect with our client. As we know, different people different opinion .so it really helps to improve our ideas.

In our survey, we asked multiple question about gardening. Many people fulfilled the survey in online. Like what are the criteria's for the selection of land , what type of flowers they want ,which services they asked from us or from the nursery owners, what will be their preferable budget and most importantly whey they want to create of a rooftop garden etc. These were all the survey question for our client.

Phase 2: Project Structure Setup

In this stage, we prepare our project structure based on the draft knowledge gained from the background research and the group discussion among our group members.

Phase 2 tasks:

Background research about the scope of urban farming

1. Gathering research articles and online resources related to the project
2. Project proposal draft
3. Introduction of the proposal writing
4. Literature review of the project writing
5. Interaction diagram design
6. Project architecture draft
7. Component design of the online platform
8. Feature list design of the components

Phase 3: Development Stage

In this stage, we intend to develop our project into several stages. In this specific project we are using incremental development strategy to build the whole application. So every time we build a feature in the development stage it must go through the deployment stage which is actually the next phase of this project. Each feature developed must undergo the deployment stage also even if the whole component is not yet to be deployable.

Phase 3 internal stages:

❖ Stage 1: Selection of necessary HW/SW requirements

In this stage, we select the necessary tools for the project as such as the SW tools for the project as well as we analyze if there are any requirement of IoT device.

❖ Stage 2: Selection of coding environments for the project

In this stage, we decide what framework we use for programming

❖ Stage 3: User interface Design

In this stage, we decide the user interface designs and develop the modules

❖ Stage 4: Business logic Implementation

In this stage, we implement the functionalities of the software

❖ Stage 5: Database Development

In this stage, we implement database for the project.

Phase 4: Deployment Stage

In this stage, we deploy our project in our test environment. After building a single component we undergo all the stages of the deployment. The list of internal phases of this stage is as follows:

- a. Integration stage
- b. Prepare the project for local application server
- c. Deployment the solution into live network
- d. Testing the project with demo users
- e. Testing the project with real users
- f. Analysis of the test data

After developing the design of the phases we created a sequence diagram for the required milestones for the project. Figure 25 demonstrates the sequence diagram of the timeline of the development of the phases of the application.

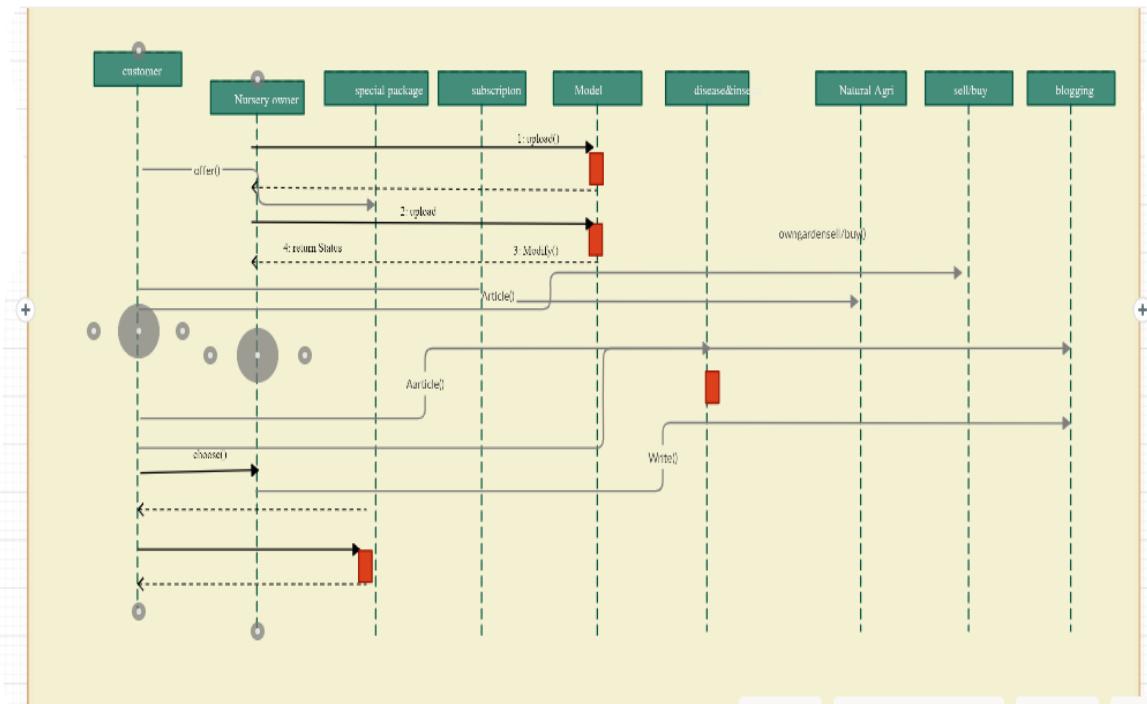


Figure 27: Sequence diagram

After designing the sequence diagrams we created a SDLC diagram for the development of the mobile application. The Figure 26 is the presentation of the SDLC diagram of the application development.

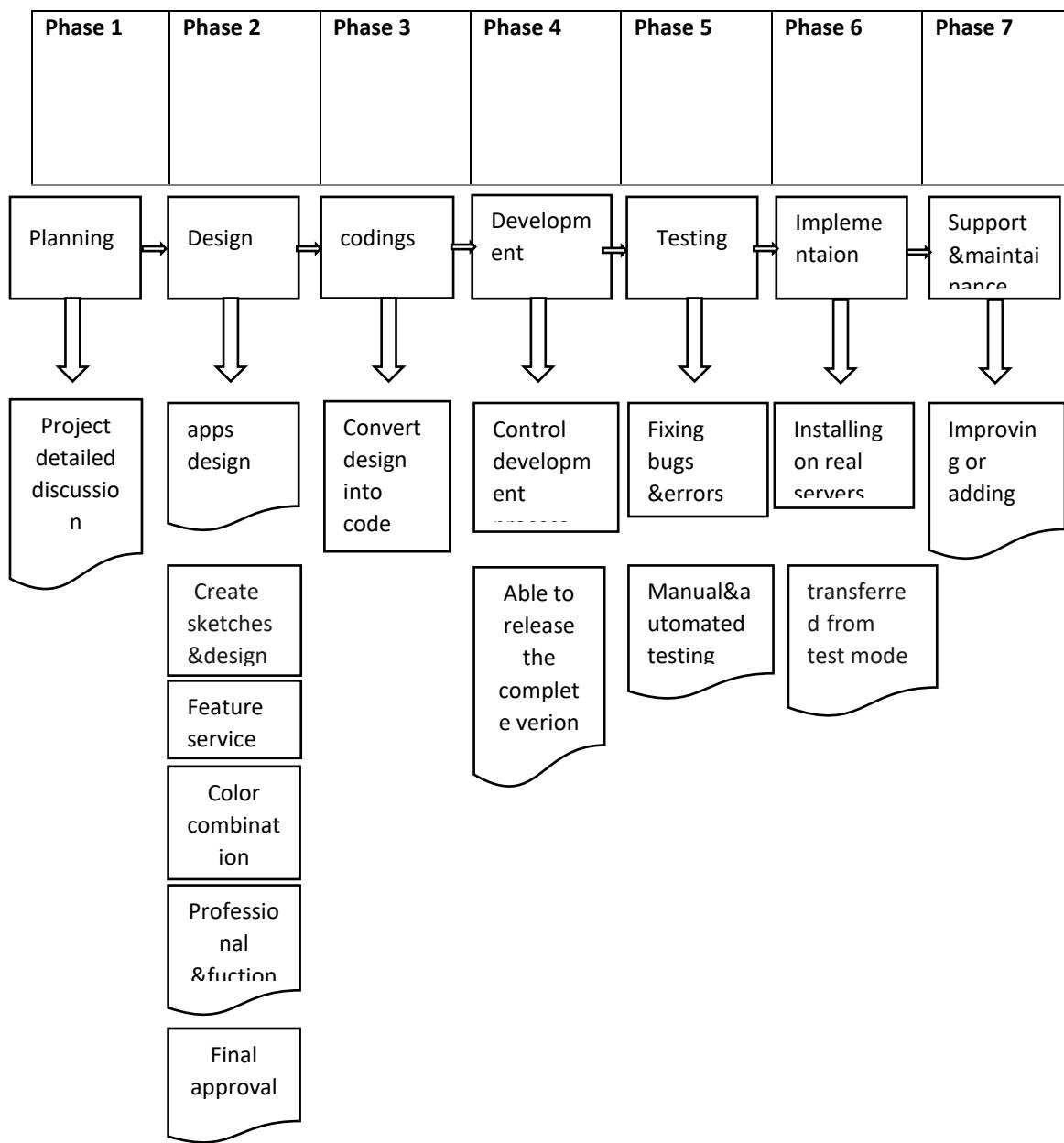


Figure 28: SDLC of Application Deployment Phase

After designing all the necessary development stages we divided our tasks between our group members the chart of the WBS is demonstrated in figure 18 for the development of our first and second prototype which was developed and tested from week 1-13 and 14-26 accordingly.

Week	Task/Activity	Members involved
1	1. Draft survey design, 2. Final Discussions of the survey,	Mubina Rahaman, Fatiha
2	1. Background research about the scope of urban farming 2. Gathering research articles and online resources related to the project 3. Project proposal draft	Mubina Rahaman, Fatiha
3	1. Introduction of the proposal writing 2. Literature review of the project writing 3. Interaction diagram design 4. Project architecture draft	Mubina Rahaman, Fatiha
4	1. Component design of the online platform 2. Project Proposal Final Version deliverable	Mubina Rahaman
5	1. Feature list design of the components a) Consumer module features b) Service Provider Features c) Source Provider Features	Mubina Rahaman
6.	Phase 3: Development Stage a. Selection of necessary HW/SW requirements b. Selection of coding environments for the project	Fatiha
7.	UX design : Basic Layouts of the projects	Fatiha
8.	UX design : Consumer Module Service provider Source Provider Admin module	Fatiha
9.	Business logic implementation: Component implementation: Customer Module	Mubina Rahaman
10.	Business logic implementation: Component implementation: Service Provider Module	Mubina Rahaman
11.	Business logic implementation: Component implementation: Source Provider Module Admin Module	Mubina Rahaman
12	Survey data analysis Modification of components and implementation	Mubina Rahaman,Fatiha
13	Deployment of component and testing	Mubina Rahaman
Week	Task/Activity	Members involved
14	Development & Implementation Of Apps(Research & Design analysis)	Fatiha, Mubina
15	Module, Interface & Program coding of apps	Fatiha, Mubina
16	Program coding	Mubina
17	Field work(Community build up with owner)	All

18	Contact with Some famous people	All
19	Connecting server	All
20	Web Application environment setup And DB import	Mubina Rahaman
21	1. Admin Middleware Development 2. DB connection class Development 3. Model Class for various operation	Mubina Rahaman
22	1. User Middleware Development 2. Necessary Controller Design 3. View template design for the application 4. View Group Set up	Mubina Rahaman
23	1. Development of the Sub-Modules 2. Development of Consumer Features 3. Development of the Source Provider Features	All
24	1. Development of the Service Provider Features	Mubina Rahaman
25	1. Authentication Middleware Development	Mubina Rahaman
26	1. Validation of the Implemented Software patterns 2. Development of the test Cases	Mubina Rahaman

Table 4: Task activity

4.2 Financial Plan and Costs:

Financially cost requirement was the most crucial part of our application development. We plan to develop the application for community building for the better and environment friendly Dhaka city. So we invested our money to build the project. Later we hope the product will generate revenue when users will use the web application. We made cost review fort the development of the application. The chart is given bellow:

Serial No.	Component Name			Cost
01.	Website Development (UX design and Development,Domain,Hosting)			35,000
02.	Local Community Build-up (user,nursery)			15,000
03.	Special Community Build-up (veterinary)			20,000
04.	Service Provider (Nursery equipment,seed,plant,medicine)			20,000
05	Monitoring And update(Developer)			10,000
06	Image processing And Data analyzing And storing			25,000
Total				1,20,000

Table 5: Financial cost diagram

CHAPTER 5

PROJECT SUMMARY

5.1 Result and Discussion:

In this project we tested our code run our application in the remote host server and also in the local server. After deploying the application into the local server we deployed our code base into a hosting server. This deployment enables the users to interact with system. Our system is fully operational right now and some developments are still going on for future releases. This section gives our through summary of our test run of the overall web application, field testing reactions of our prototype, the performance review of the system and the errors of the application.

5.1.1 Test-Run of the Application

Our web application at first is tested on our local server. The homepage of our system is presented in Figure 27. Here as we can see the logo and the necessary info graphic is presented.

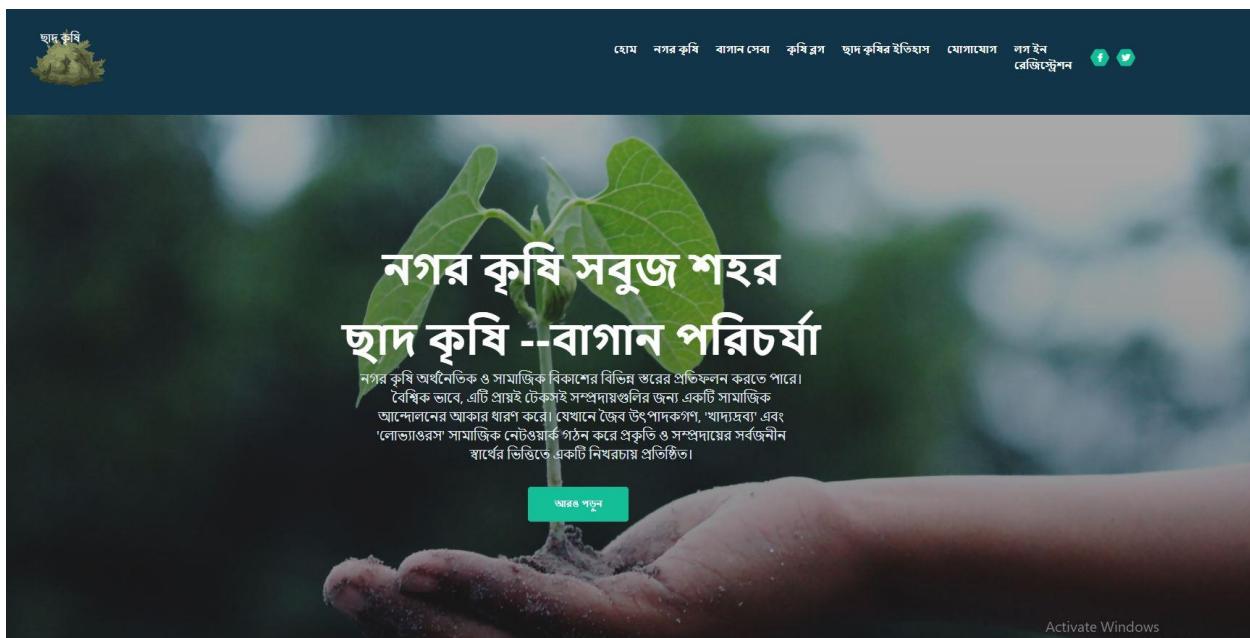


Figure 28: Home Page of Urban Faming Application



চুমকি

আমরা সবুজ ঢাকা গড়ার পথে উদ্যোগী

ঢাকা শহরে ক্রমপক্ষে প্রায় সাড়ে চার লক্ষ ছাদ রয়েছে (সাড়ে চার হাজার হেক্টেরে বেশি) যা দেশের কেন্দ্র একটি উপজেলার সমান বা বেশি। মেখানে বাসযান, ফ্ল্যাট-কলেজ, বিশ্ববিদ্যালয়, অফিস-আদালত, ব্যাংক, শপিং মল, কন্ডোনেশন সেন্টার ইত্যাদি বেশি। বিভিন্ন কোণেতে ২০% সবুজ থাকার কথা ধার্কলও মানুর বিষয়টি আশ্চর্যজনক নয়।

শহরবাসীর পষ্টির বিষয়টি মাঝায় নিলে ফল ঢাক

অত্তোরশ্চকীয় বিষয় হচ্ছে সরবতি, ফুল ও সৌন্দর্যবর্ধনকারী গাছ-পালা লাগানোর পরিমাণ শহরে তুলনামূলক বেশি। এটিৱে অশুর বিষয় যে, নগরে সবুজের পরিমাণ বাড়ুচ্ছ কিন্তু এর সাথে সাথে পুষ্টি পাওয়ার বিষয়টি গুরুত। সহকারে ভাবতে হবে এবং তা এখনই করার উপযুক্ত সময়। একবিংশ শতাব্দীতে আজো হত নগরায়নের এবং জনসংখ্যা ২০৫০ সালে ৮.৩ বিলিয়ন ছাড়িয়ে যাব। বর্তমানে পুরুষীর ৩৮% ছল প্রায় ৮০০ মিলিয়ন হেক্টর কৃষি। এবং আঙ্গোল এবং অধিবিক্ষু জনগণকে খাওয়ায়ের জন্য এখনকার কৃষি পদ্ধতি ব্যবহার করলে আজো ১০০ মিলিয়ন হেক্টর কৃষি জমি প্রয়োজন পড়ব। কিন্তু এই পরিমাণ কৃষি জমি আবশিষ্ট নাই যার ফলে বাণিজ্যিক ভাট্টিকাল চারিবাদ ছাড়া অন্য কোনো বিকল নেই যা অবশ্যই নগর কৃষির অন্তর্গত। গবেষকদের ধরণা অনুযায়ী, সুষ্ঠুচ্ছ ৩০তালা ভবনে ২৫,৮০০,০০০ বগমিটার ভাট্টিক্যাল ফার্ম স্থাপন করা যায় যা নিম্নে ৫০,০০০



Figure 29: Blog Page of Urban Faming Application



লগ ইন

ফোন নাম্বার

আপনার পছন্দের পিন প্রবেশ
কোড

Remember Me

লগ ইন পাসওয়ার্ড ভুলে গেছেন?



Figure 30: Authentication Page of Urban Faming Application

রেজিস্ট্রেশন

আপনার নাম	Md Jahidul Haque
ইমেইল	sd
ফোন নং	<p>Please include an '@' in the email address. 'sd' is missing an '@'.</p>
আপনার সম্পর্কে বলুন	নাস্রিয়া মালিক
আপনার পছন্দের পিন প্রবেশ করুন
আপনার পছন্দের পিন কনফার্ম করুন
Register	

Figure 31: Custom Validation of Urban Farming Application

In Figure 29 and 30 we present the front end of the system which is liked by most of our users. We used Bangla as the primary front of the system. In the Figure. 31-33 we showed the authentication UI and custom validation during the registration process. Our algorithm lets the user to verify their email address via correct input and unique email address. In Figure 34 and we presented the Provider portal and admin module of the system where user can user variety of services.

রেজিস্ট্রেশন

আপনার নাম	<input type="text" value="Jerin Rahaman"/>
ইমেইল	<input type="text" value="jerin@gmail.com"/> ×
The email has already been taken.	
ফোন নাম্বার	<input type="text" value="12344"/>
আপনার সম্পর্কে বলুন	<input type="text" value="বাগানে আগ্রহী"/>
আপনার পছন্দের পিন প্রবেশ করুন	<input type="text"/>
আপনার পছন্দের পিন কনফার্ম করুন	<input type="text"/>
Register	

Figure 32: Custom Validation of Urban Faming Application

Figure 33: Nursery Portal of Urban Faming Application

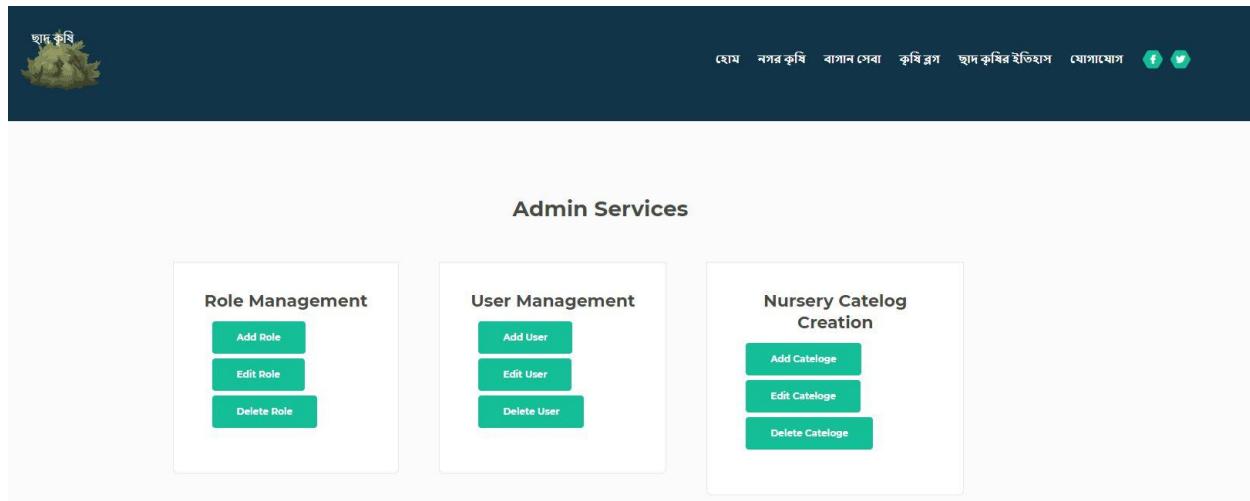


Figure 34: Nursery Portal of Urban Faming Application

5.1.2 Field testing Reactions on the prototype (Experimental Data Analysis)

After the development and the deployment of the application we physically visited to the nurseries to show the nursery owners and the basic users who want to grow their plants. We got the following responses for the survey presented in Figure 34,35, and 36

6. If you are offered a service where service provider will do the daily maintenance(watering plants, spreading fertilizers and pesticides, overall maintenance) for your garden, how much would you like to pay monthly for that service?

15 responses

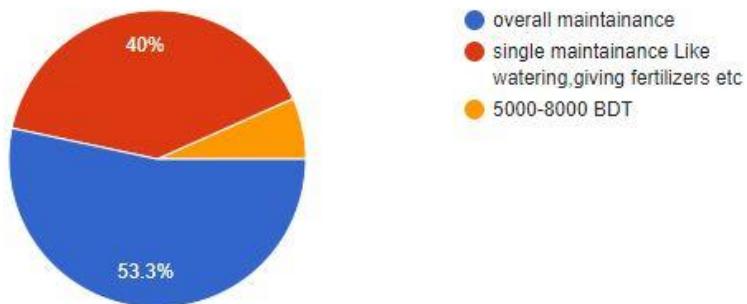


Figure 34: Response regarding services of Urban Faming Application

5.If you are offered a service provider will setup the garden(provide seeds,soil,fertilizer,initial maintenance) for you,how much would you like to pay for that service?

15 responses

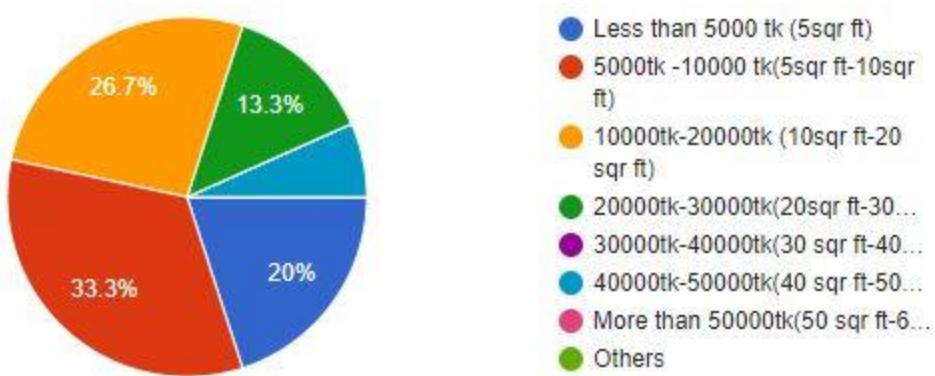


Figure 36: Response regarding service cost of Urban Faming Application

3.What types of plants you want to grow for gardening?(You can choose multiple options)

15 responses

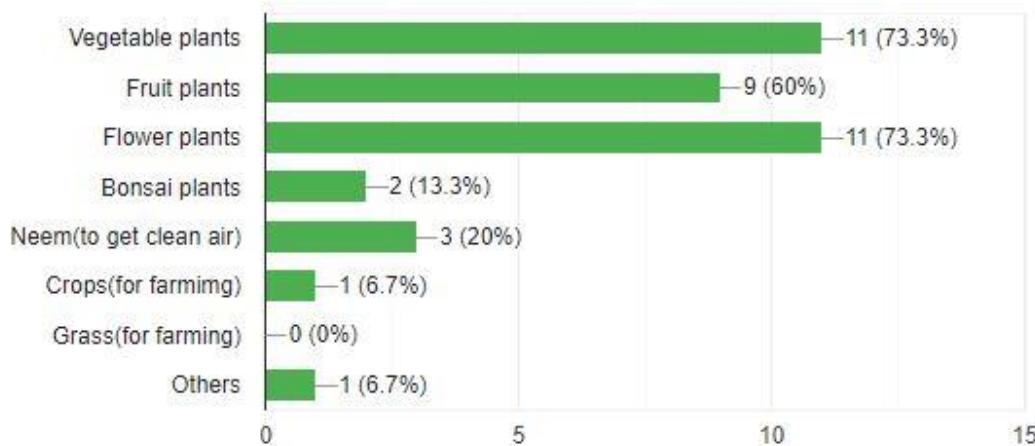


Figure 37: Response regarding Plant type choice of Urban Faming Application

After reviewing the responses we came to the conclusion that city dwellers wants to grow their plants due to the need of fresh vegetables, fruit plants and for the sake of the beautification of their urban spaces. Most of the users wants to spend fair amount of money to grow their garden.

Then we physically visited several nurseries to know the status of the nursery business. After the visit, we came to know that most of the nurseries are ran by the caretaker who has enough knowledge of gardening. The owner of the nurseries only monitor the caretakers. So, the implementation of service and source provider module in our first prototype needs to be replaced for the better usage of the system.

Then we redesigned our system with new Provider module where user can only manage their nursery inventory. The nursery owner can take photos of their plant and can upload the information about all the plant and garden selling items into the inventory management of the system. Then they can categorize the information based on price and post in the system to sell. Furthermore, they can monitor the tracking of any of the order of the customer.

5.1.3 Errors of the Existing Systems

Our system has several errors which we found during the deployment stage and UX testing phase. We observed problem during order tracking and order management. When a user orders a plant to buy the following 2 steps happen:

- a. Call the nursery caretaker to get the plant/other staff
- b. After that nursery owner update all their information on orders at days end.

This step creates problem sometimes so that user sometimes doesn't get update new information about the order.

The deplorable the truth is that corner cases are important for working together in the advanced world. To add salt to the injury, it's frequently your best clients who wind up discovering them, on the grounds that the force clients of an application will run into its constraints and characteristics before a typical client.

Fortunately, this presents a brilliant occasion to convey an extraordinary client experience for those clients! Tune in to input from your clients. In the event that they're pushing your application as far as possible and hitting edge cases, work with your designing group to arrive at an answer which will assist them with taking advantage of your app. If they are squeezing an unusual blend of catches to trigger an accident, set aside the effort to comprehend why they picked that mix. It may prompt you changing some befuddling UI and make a superior application experience for everybody. The best applications figure out how to nimbly deal with an edge or corner case, and picking an incredible improvement group will be the initial step you can take to ensure your clients capitalize on your application, regardless of how they use it.

We can solve most of the corner cases by making smart model decisions up front. To find a solution for a corner cases at the earliest step of improvement is to use believe open source software for common component for our app.

5.1.4 Test run of the mobile application

The Figure 36 and 37 present different pages of the mobile application. The Figure 37 presents the blog list page of the application which is plant doctor feature of the app. People can see the different diseases related info of plants.

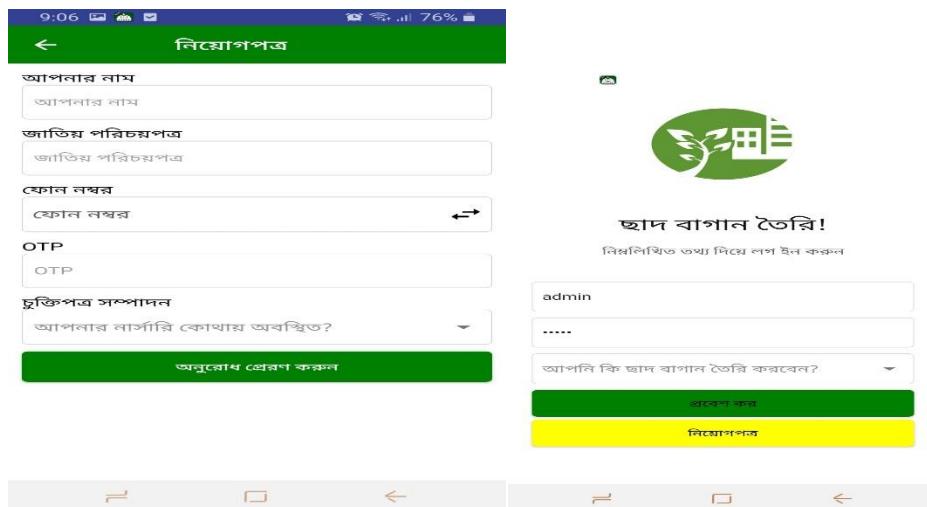


Figure 37: Registration and login page of Mobile Application

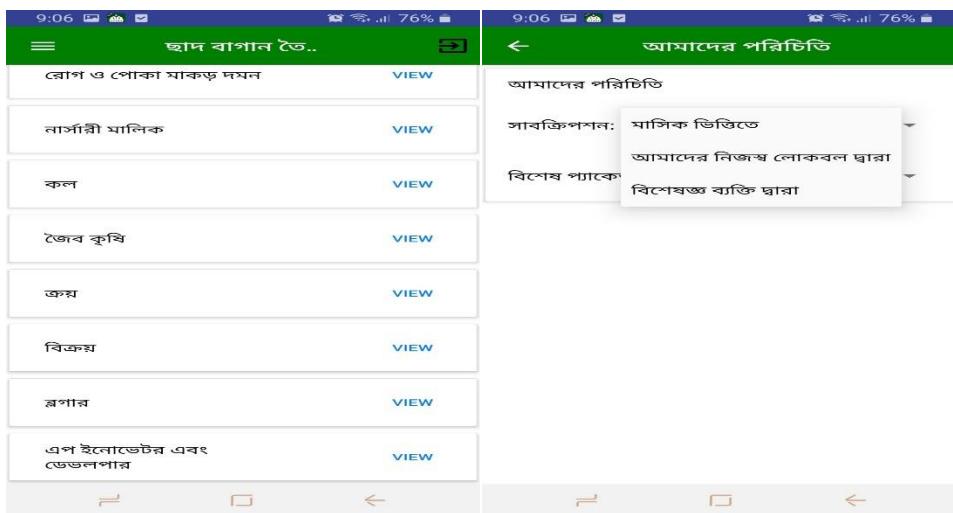


Figure 38: Blog View pages of Mobile Application

5.2 Feasibility Study:

The garden application provides the online platform where different agents and actors plays together to make possible the rooftop gardening concept among the city dwellers in cities, primarily in Dhaka City. In Dhaka, the nursery owners sale their plants and catalog them

manually. They have no access to the database of the consumers and cannot sale their product via online. The online platform for the plantation can be incremental for nursery owners to sale their products as well as the consumers can buy their products. We prepared a financial plan to make the product financially viable.

Financial plan of the system

There will be available three packages for customer .

- a. Free Article :
- b. Where customer will find lots of article regarding trees, their problem , how to take care garden, trees etc.
- c. Image process system
- d. Consultancy

In Consultancy package, we have added various type of offer for customer.

Package 1: Garden setup and design

Some design picture will be available in our app . From there customer can select a significant one from their interest.

Package 2: Man power (Nursery people)

We will supply people for setting up the garden.

Package 3: Subscription (per month service)

In this package, there will two option. 1. Gardener & 2.expert service. These two packages will be available for one time or per monthly as per customer's need.

Package 4: Selling & buyer

People can buy and sell their own garden's vegetable, fruit.

SMART PACKAGE:

- Aqua Farm
- Hidro Farm
- Drop water irrigation /gardening
- Green Housing
- Smart Gardening & Monitoring

Both users will have an account. Who needs help or who are going to help.

We want to build up a community between these people where they can share their thoughts, their garden, their emotions about garden.

The Viability of the application in terms of target users

**** Users who want to grow trees:** For the customers who don't know not enough about urban gardening or urban farming. The urban gardening application is definitely a solution due to bellow features:

1. **Urban plant info:** Plant info is a feature where a guest user or a registered customer can use to gain insight about different types of plants based on their category.
2. **Urban plant search:** Plant search is a search tool were customer can view a particular plant and can gain knowledge about cultivation of the plant.
3. **Urban gardeners info:** When customer have enough knowledge about rooftop gardening he/she can definite use this feature to build their own garden or get help from other gardeners in terms of tips/gardening-service.
4. **Service ordering:** Using this feature use can easily choose a gardener to provide him/her a full-time or part-time service for a specific time period. Using this feature customer can able to grow a new urban farm with the help of others.

5. **Material ordering:** Using this feature customer who have enough precursor knowledge on farming can order any material related to gardening.

**** Users who want to sale their plants:** Nursery owners are the primary user of the application for sourcing service or gardening material. The usability of the application for this target group are as follows:

1. **Source Inventory:** Using this feature nursery owner can manage their own inventory of items related to farming. They can add/update/delete/edit any inventory information using this feature and can also manage the orders of the customers.
2. **Service management:** Using these feature nursery owners can create a service group with manpower who can deliver different gardening services to the registered client of the nursery owners.
3. **Order tracking:** Using these feature nursery owners can get the updates of the service orders placed by the customers.
4. **Customer management:** Using this feature user can communicate.
5. **Revenue Management:** Plant seller can see the revenue generated from the plant business can manage the income by distributing it to the service workers of his/her own.

Inspecting the sustainability of the online platform

In the proposed solution of the online platform of the gardening application, we tried to achieve the goal to make awareness among people to build a greener city area. In order to do the public awareness we propose a peer to peer review system via blogging. Moreover, the online platform offer both consumer and producer end products which can be very promising for the urban gardening growth. The online platform compounds the possibility of generation

of revenue among the freelancers who wants to join this commercial viable and environment friendly cause.

Short Term Effects of the Gardening Application:

1. Online platform creates more opportunity to the nursery owners to sale their products via online.
2. Online platform creates opportunity for the unemployment people to assist both the nursery owner as well as the consumers.
3. User can now order any garden material via online application.
4. Gardening provider can generate short term revenue from the application.

Long-term Effects of the gardening application:

1. Public awareness regarding plantation in urban spaces.
2. Building a community who are concern about environment.
3. Level of oxygen in the environment will increase due to the revolution of urban plantation.
4. Healthy and cleaner urban city.

Apart from the technical issues of the project, there are lots of non-technical issues which are highlighted as follows:

- Community build up using the online platform:

Our community has the these following issues regarding the plantation in urban spaces:

1. House owner have no easy access to the nursery owners.
2. House owners have not enough sources for setting up their own urban farms in their urban spaces.

3. House owner lacks knowledge about rooftop plantation.
4. House owner want fresh air, fresh vegetable, fresh fruits and beautiful rooftop urban garden.
5. House owner wants to invest their time or money to get healthy life in their urban spaces.

In order to mitigate these above issues our online app could be an incremental platform. Our app could do the following in the aid of house owners who have private urban spaces:

1. Web app can give motivational info regarding the positive effects of urban plantation.
2. Web app can give necessary cultivation info regarding various plants.
3. Web app can give services for house owner which can be delivered by nursery owner.
4. Web app can give plantation services via third party vendors.

➤ Social Awareness using the online platform:

Gathering social awareness is a great deal of non-technical issue since it's very important for the wellbeing of our city dwellers. The following things are the challenges:

1. Socially, urban farming/rooftop gardening has to be supported by the authority.
2. People should know the benefits of urban farming in terms of healthy diet and economic betterment.

3. People should talk and share each other's thought regarding the urban plantation.
4. People can work voluntarily to help others to get success for better urban spaces.

To solve these above problems, our app could do help our society in the following ways:

1. Web app can help people to get their own income source via growing their own crops.
2. Web app can help the authority by proving services to make our city greener. Via this, authority can be benefited.
3. Web app can be used as a social blog to create awareness about planting trees in the rooftop area.
4. Web app can help free and unemployed people by giving jobs via providing third party job opening for giving gardening service to the house owners.

5.3 Problem Faced and Solutions:

After the development of our application we faced many problems. At first when we built our first prototype we don't know what are the main needs of the users who want to grow their plants in their urban spaces. Then we got learned about their need via the survey. After that we redesigned the software based on new requirements. Then we faced problem in nursery owner module testing since most of the nursery owners physically

present in their business places. So we also solved that problem by diving the order processing via manual and online process in 2 steps.

5.4 Future Development:

The online urban farming app can be developed more in future if we focus on more on the aspect of users selling their own harvested foods in the online platform. In future the following services can be added to the system:

-  Customer grown food selling platform
-  Income management for the providers
-  Customer can create new nursery business

In future we hope we will develop more of our online product and try to build the social awareness for urban gardening.

5.5 Conclusion:

Online platform for urban farming can be the potential application for creating green and beautiful Dhaka city. It can certainly help the end users with both economically and environmentally. After getting the good reviews from the customers we concluded our project as good online platform for building the community of the garden lovers. Here Garden lovers can get encouragement to setup their nursery business as a nursery owner. Apart from that customer now can order new plants or any plant accessories via online application. It helps them to concentrate their garden development more easily. Moreover, the software certainly created a bridge between the customers and the nursery

owners. Now also can the customers can get new information about expert advices from home to grow their garden more effectively. It can reduce the overall cost of the maintains of their garden. Most interestingly, our app is economically good option for the both nursery owners and the customers. Since the nursery owners wants to promote their business, they can now use the online application and get profited. On the other hand, customers now can get home services via the online platform so that its also profitable for the home dwellers. In future, if we focus more on community building the software will have certain economic success.

5.6 Project Demonstration Review:

How do you rate the registration and login service?

30 responses

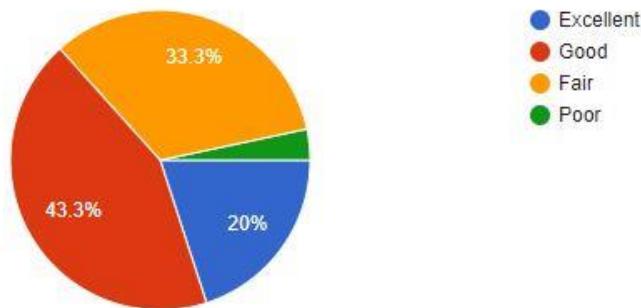


Figure 40: User response on login and registration

What is the most exciting service in the application?

30 responses



Figure 41: User response on services

What is the most user-friendly service in the Nursery portal?

30 responses

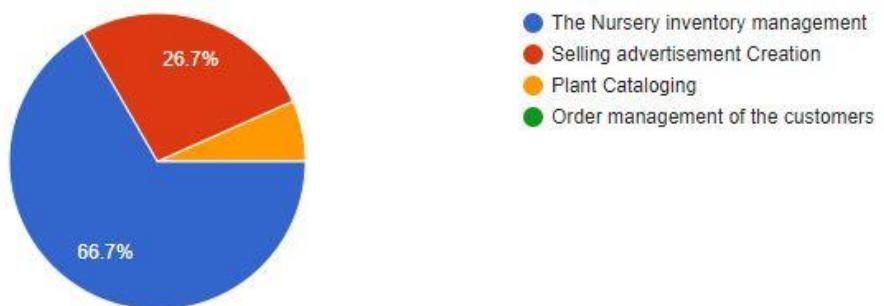


Figure 42: User response on Nursery Portal

How you rate the user-friendliness of the web application?

30 responses

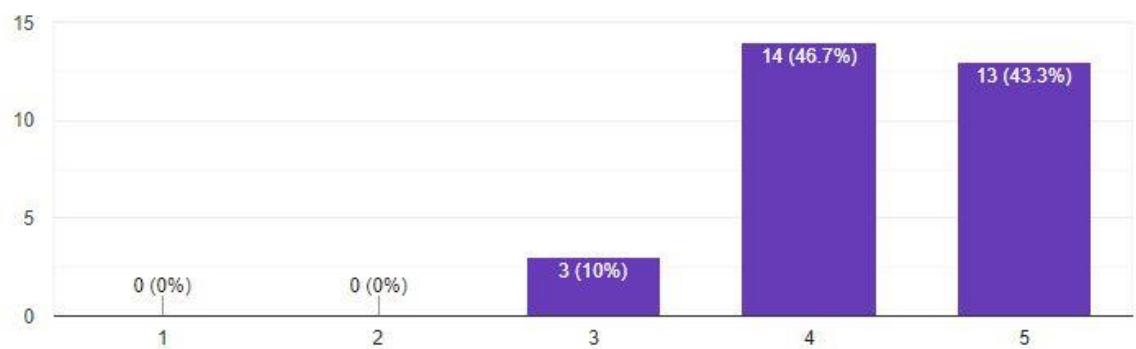


Figure 43: User response on User Friendly

5.7 Brochure:

Page 1:

The project aims to develop an online buying and selling platform of gardening material and harvests via 3rd party integration. 3rd parties include the nursery owners who wants to sell their gardening products, the individual customers who wants to sell their gardening harvests.

Urban Gardening Platform

```

graph TD
    UGP[Urban Gardening Platform] --> WebApp[Web App]
    UGP --> MobileApp[Mobile App]
  
```

In our project, we have built an android apps and web application. Through this app and website, we will build up a relationship between Nursery owner and Green lovers. Not only that, we have a feature where people can sell their own fresh vegetables. Our main goal is to build up a communication between customer and nursery owner. This is a special additional feature.

For Home dwellers:
Now can you buy your Garden materials from home. Just order and call nurseries.

Plant doctor:
Now can get help from experts. Now garden owners under one roof. Using the online blog of expert advices, you can get advices to cure the diseases of your plant.

Track your order:
Now get tracking your parcel yet to come and use the garden items.

WRITE YOUR BLOG AND BE AN EXPERT:
Now User can also write their blogs and share with other gardeners.

For Nursery owners:

Service Creation:
Using this feature a provider creates list of packages were each package has list of services. Provider can also set the package cost using a form

Warehouse Management:
Now you can manage and add plants in your own nursery warehouse

Manage your Ad:
Now Post the plant items and get attention of the buyers!

MANAGE ORDERS:
Now keep track of your orders.

Page 2:

Our Architecture

FIGURE: MOBILE APP ARCHITECTURE

Mobile APP Design Flow

Our UI

Conclusion

Although we tried to implement the most favorable features for the urban rooftop farming, some important updates still need to be developed in future. An online platform can easily grow a lot via the online communication and people can really get help by getting information via the application.

5.8 Poster:



Online Urban Farming: Introducing online platform for garden lovers

Mubina Rahaman Jerin and Fatiha Jahan
Department of Electronics & Computer Engineering, North South University

Motivation and Objectives

Urban agriculture is the one of the most relevant environmental innovation of the modern era. Sustaining green habitation in city areas has a great social, environmental and commercial effect among the city dwellers. In fact, the decision of urban plantation and rooftop gardening in city can make overpopulated city area more suitable for living with greener environment. Implementation of innovative ideas in this domain can increase the chance of a healthy environment. IT products can certainly be a sustainable solution to network people in the noble cause to make their populated cities with more green habitation. In this project, we aim to design, implement and test a software based solution for buying, selling, networking and giving knowledge about urban agricultural products among the online users.

Summary of the problems: Dhaka City suffers a lot due to air pollution. For this we need to best utilize the urban spaces by setting up green habitation more and more. To pace the growth of urban agriculture. We need IT solutions.

Aim of the research: Solving Urban Agriculture equipment shortage by offering a online platform for all.

Approached methods: We Implemented A web Platform and Mobile application for solving the lack of communication of users who wants to grow their garden but have not enough information and material support for gardening.

Urban farming Solutions

Urban Farming Web-APP VS existing Apps

Website Technology	Main Features of the existing system	Available Features in Urban Farming App
React Native	Main Features of the existing system	Available Features in Urban Farming App
Flutter	Shopping of garden kits, Garden Community	Here also this feature is available
React Native	Membership based service for gardening	Now also care for membership based garden
React Native	Education services, Online shopping, Online Payment, Email	Offer the services that user want, Right now we are not able to provide, If different options will be available then we will provide space to put those in the application
React Native	\$100 garden designing	Smart services of urban farming is not considered in the existing system
React Native	\$100 of 100 devices for urban agriculture can be considered	Smart services of urban farming is not considered in the existing system

List of android apps on online Rooftop farming

- Rooftop Garden Ideas : In the apps, lots of picture are available .
- How to make a Garden : Details of the basic gardening like what types of trees, flowers can be planted , how to take care , scientific way etc.
- Roof Gardening Bangla : How to create a roof garden . What to do, vegetables growing , various tips about gardening etc.
- Chud Krishi : Rooftop gardening , what can be done , what are the process. Fruits & Flowers growing process .

Key Features of Urban Farm

OUR MOTO

“বেশী করে গাছ লাগান, ভালো থাকুন”

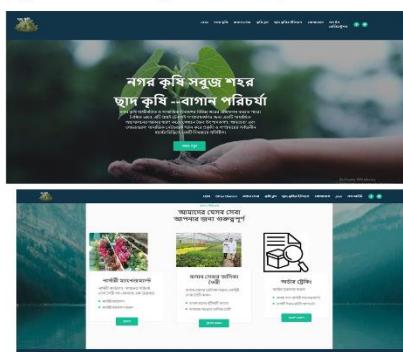
For Home dwellers:

Online Buying at your home: Now can you buy your Garden materials from home. Just order and call nurseries.

Plant doctor: Now can get help from experts. Now garden owners under one roof. Using the online blog of expert advices you can get advices to cure the diseases of your plant.

Track your order: Now get tracking your parcel yet to come and use the garden items

Write your Blog and be an expert: Now user can also share with other garden lovers about the topics on urban farming.



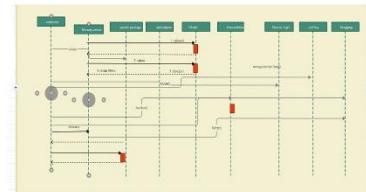
For Nursery owners:

Nursery Inventory: Using this feature a provider creates list of inventories of different types of plant or garden materials.

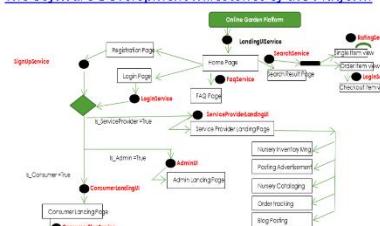
Plant Management: Using this feature Nursery Owner can put new information in the inventory list.

Service Creator: Using this feature a provider creates list of packages were each package has list of services. Provider can also set the package cost using a form.

The Software Design Methods



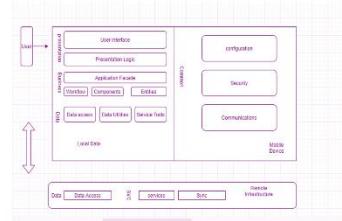
The Software Development Milestones of the Platform



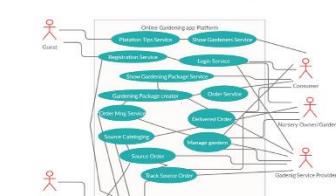
The Interaction Diagram of the web application

Module Architecture

The System Level Design Of Mobile App



The UML Design of the web application

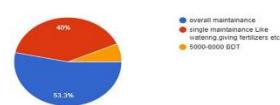


Impacts of the application

After the development and the deployment of the application we physically visited to the nurseries to show the nursery owners and the basic users who want to grow their plants. We got the following responses for the survey presented in the following Figures :

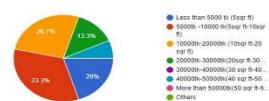
6.If you are offered a service where service provider will do the daily maintenance(watering plants,spreading fertilizers and pesticides,overall maintenance) for your garden,how much would you like to pay monthly for that service?

15 responses



6.If you are offered a service provider will setup the garden(provide seeds,soil,fertilizer,initial maintenance) for you,how much would you like to pay for that service?

15 responses



At last we had a understanding the application should be build in terms of the flexibility of the Nursery Owners. Because Many of them have no mobile phone.

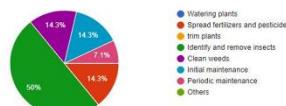
After math of the final Prototype

After building and the deployment of the final prototype we came up with following scenarios:

1.We need to build our application by considering our application

4.Which service you want to get?

14 responses



more focused on diseases of plants. 50% of the respondent of the application wants the service for getting tips regarding the diseases of the plants. Our Mobile App can give advices in this

Implemented Feature for this:



Conclusion

Although we tried to implement the most favorable features for the urban rooftop farming, some important updates still need to be developed in future. A online platform can easily grow a lot via the online communication and people can really get help by getting information via the application.

5.9 Project Demonstration

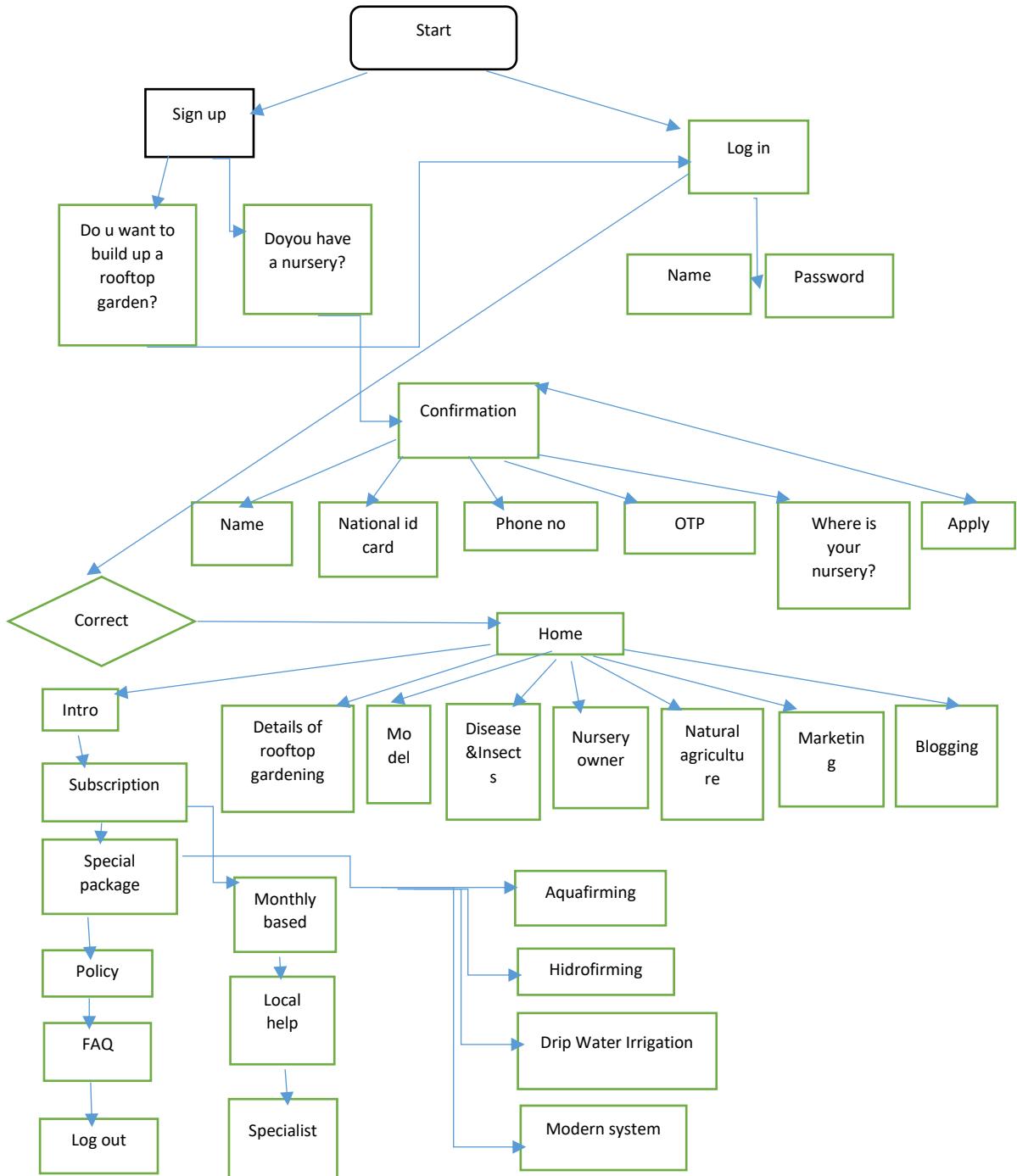


Figure 44: Code Flow Urban Farming Application

Project Review and Responses:

How do you rate the registration and login service?

30 responses

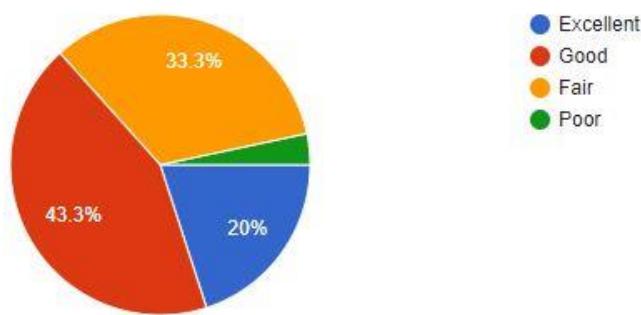


Figure 40: User response on login and registration

What is the most exciting service in the application?

30 responses

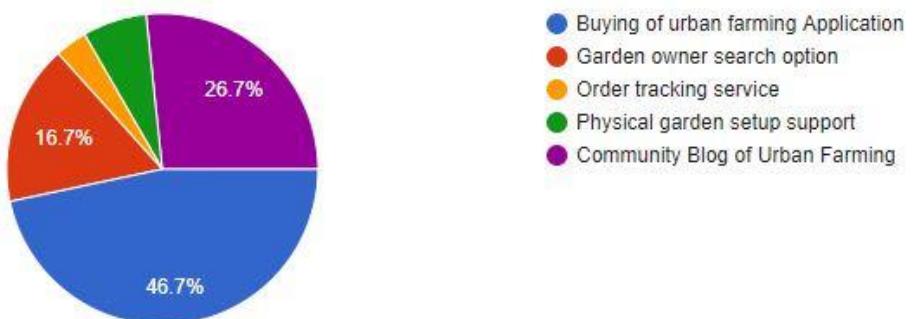


Figure 41: User response on services

What is the most user-friendly service in the Nursery portal?

30 responses

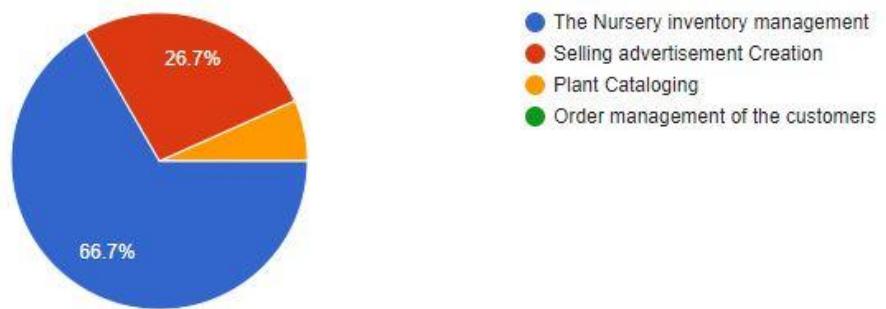


Figure 42: User response on Nursery Portal

How you rate the user-friendliness of the web application?

30 responses

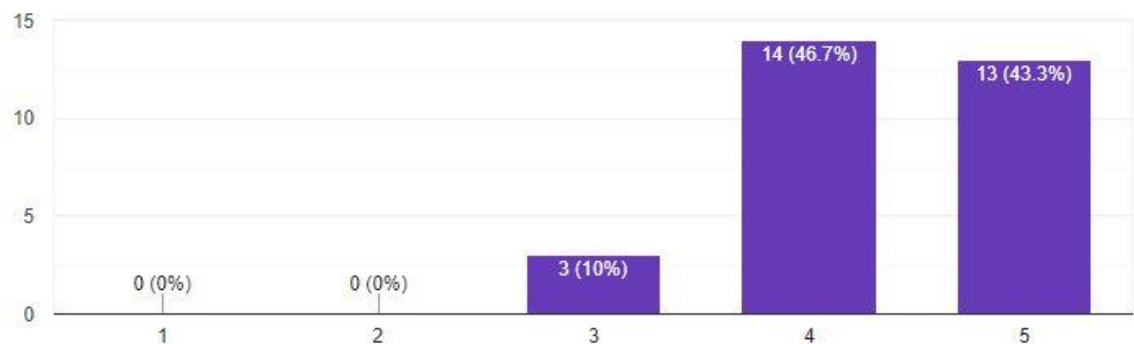


Figure 43: User response on User Friendly

5.10 IEEE format paper:

Page 1:

Online Urban Farming: Introducing online platform for garden lovers

Mohammad Rezaul Islam
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
mubina.jerin@northsouth.edu

Mubina Rhaman Jerin
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
mubina.jerin@northsouth.edu

Fatiha Jahan
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
tanwiththebird@gmail.com

Abstract—

Urban agriculture is the one of the most relevant environmental innovation of the modern era. Sustaining green habitation in city areas has a great social, environmental and commercial effect among the city dwellers. In fact, the decision of urban plantation and rooftop gardening in city can make overpopulated city area more suitable for living with greener environment. Implementation of innovative ideas in this domain can increase the chance of a healthy environment. IT products can certainly be a sustainable solution to network people in the noble cause to make their populated cities with more green habitation. Moreover, online software solution can ease the process of urban plantation in urban spaces like rooftops, balcony, enclosed spaces with little sunlight. More specifically acquiring the tools and services related to gardening is seldom in the city. Online software-based products can help people to network to other people who have implemented knowledge about urban farming. Also, people can get to buy and sell agricultural tools as well as the harvest. In this project, we aim to design, implement and test a software-based solution for buying, selling, networking and giving knowledge about urban agricultural products among the online users.

Keywords—*Urban Farming, Mvc Framework, Rooftop farming, Online application, Urban gardening*

I. INTRODUCTION

In the modern era of science and technology, buying and selling of goods in ecommerce application became increasingly popular. Nowadays, almost everything is possible to get via online. In the context of Bangladesh, agriculture is the one of the most crucial sector for the growth of our middle income economy. Most of the developed cities in our country are densely populated. Even most of the cities are suffering rapid unplanned urbanization. This cause destruction of green habitation inside the city area. But green habitation is very important for maintaining a healthy air inside the city. Due to organizational factors, often city areas are deemed to have more commercial values and it caused no green habitation. In the context of Dhaka, roughly 3% of the area is covered by green habitation. It is a very low ratio of green habitation in terms of land area when we compare it with most developed cities all around the world .

Online platform for ecommerce of agricultural products is really crucial for developing interest in urban farming among the city dwellers. Using the ecommerce site users can buy and sell garden products and also users can gain knowledge via sharing the tips regarding the gardening. Furthermore, logistics services via online platform can benefit the sales of the nursery owners who have little outreach of the customers. Online services of the ecommerce solution of the urban farming can consists usually delivering the gardening tools, seeds, manpower, 3D plans of the gardening, tips, and experimental designs.

II. BACKGROUND LITERATURE

A. Motivation in Online Urban Farming

Urban gardening or urban farming could be the perfect solution for reducing the overwhelming pollution inside Dhaka city. Urban farming is the cultivation of food within metropolitan cores as opposed to that in more pre-urban and rural areas [1,2,3]. Bos presented the impact of social media for creating urban farming as a sustainable food network in city areas [4]. Urban green spaces or greeneries are predominantly crucial for proper functioning of the ecosystem in any urban environment [5]. Urban green spaces can play critical role in conserving biodiversity, protecting water resources, improving microclimate, sequestering carbon, and even supplying a portion of the fresh food consumed by urban dwellers [6]. Author in [7] asserted that the green resources in Dhaka are overwhelmed by a number of limitation inherent to the exceptional mode of urban development, rapid increase of urban population, transformation of green and other open spaces into other types of land use, lack of proper planning and implementation and management restrictions as well [4]. Socially acceptable urban agriculture businesses: this paper is suggesting that we can take this urban rooftop farming as business purpose [8]. Brooklyn Grange is the world's leading soil rooftop farming company and sustainability center for providing services to its customers a platform to get knowledge about urban farming and services of plantation confirmation on urban gardening. But the online service of the company can't be used in Bangladesh. The importance of local online services is really crucial for spreading the urban farming in Bangladesh.

B. Existing Solution for Urban Gardening

There are several apps with local and international affiliation can address in the table 1 where we identified the basic mode of business and features of the applications.

There are many online application is available on google Playstore. We have searched on Playstore and found a various types of online based rooftop gardening apps.

1. **Rooftop Garden Ideas** –In this apps, client will find only various types of design.as we know, Different people different interest .so people can find rooftop gardening design based on their taste, personality.

2. **How to make a garden** –This apps has the details of the basic knowledge of gardening .choosing the right soil ,how to make tub ,how to grow plant ,what's need to done for taking care of, scientific way etc.

3. **Roof gardening Bangla**-how to create a roof garden, basic knowledge of growing plants, tips about gardening.

Website/Tech.	Category of Business	Features
Nagarkrishi	Retail sale of own garden kits	Shopping of garden kits
Prkriti Farming	Garden Services	Membership based service for garden management
Ugao	Ecommerce	Garden services, online shopping of garden kits
Smart Garden	3D garden designing	3D design of garden space.

TABLE I. Existing Solutions

III. TECHNICAL DETAILS OF THE APPLICATION

The urban gardening application is an application to provide gardening services to the customers who own the urban spaces. This application is a bridge between two communities: the plantation service providers who lack technological support and the city dwellers who want to grow garden in their urban spaces.

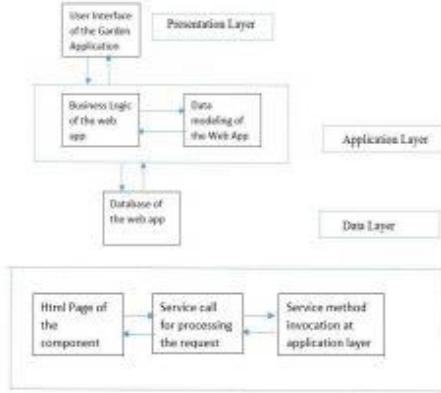


Fig. I : Top Down View of the Application

We build separate web application and mobile application to facilitate urban gardening. More architectural design of the application in the following sub-sections.

A. System Level Design of the Web application

System level design of the web application consists of different technological parts. We implemented our application in Laravel application framework. Our Application consists of different modules with range of services. Figure 2 shows that:

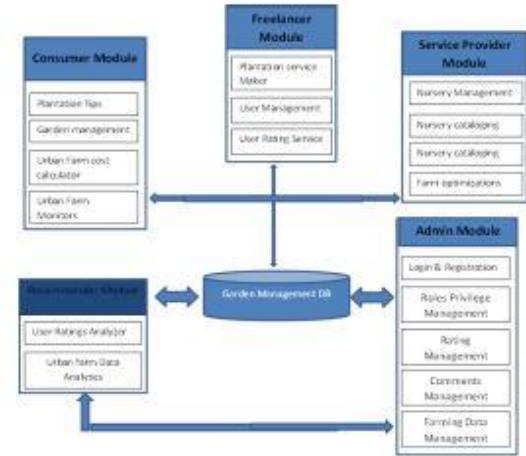


Fig 2. System Design of the Web Application

B. System Level design of the Mobile Application

The mobile app has the system level architecture presented in the Figure 3.

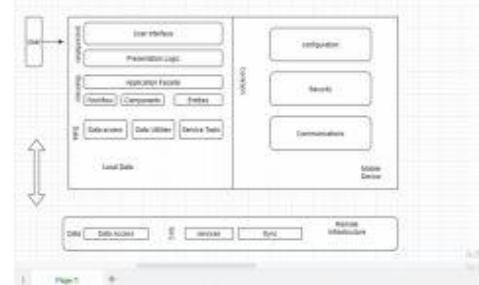


Fig 2. System Design of the Mobile Application

C. Module Descriptions Of the Web Application

Our newly developed final prototype has four basic modules: For the web application:

1. Admin Module:

This is the module where server/app administrator manages the user roles, privileges, and other administrative tasks of the web application

2. User Module:

This is the module which is used by 3 types of user category: Customer, Nursery Owner/Gardener, Service Provider

3. Authentication Module:

This module helps the users or administrator to validate the form data or other data driven functions.

4. Data Management module:

This module helps the end users to manage their information also this module can predict/suggest new information via its data visualization capability.

Page 3:

D. Module Descriptions Of the Mobile Application

. There are a couple of highlights which are ideal to have and some which are unquestionable requirements, yet many will make your application truly stick out. By keeping all these fact in our apps so that people don't feel bored using our apps. People don't like complicated thing especially some things what they will use in their regular lives. Apps should be very easy to use, need to fulfil some basic needs of clients.

So, we have created some feature for our apps for our clients.

1.Modesty: Numerous individuals have limited capacity to focus. So if you make your application hard to explore or complicated and it's very tough to understand at that point they will lose their interest.

2.Speed: Quick stacking screens are essential. Nobody likes pausing, especially when all they need something to know or looking for something and they see this at is a screen loading and this is very disappointing and irritating for a user. If they need to wait this long, they will look for something else which is better than this. Speed implies a proper arrangement of illustrations and not bringing enormous tables and information bases. Keep it basic and keep it fast.

IV. RESULTS AND DISSCUSSION

In this project we tested our code run our application in the remote host server and also in the local server. After deploying the application into the local server we deployed our code base into a hosting server. This deployment enables the users to interact with system. Our system is fully operational right now and some developments are still going on for future releases. This section gives our through summary of our test run of the overall web application, field testing reactions of our prototype, the performance review of the system and the errors of the application.

A. Test-Run of the Applications



Fig. 1. Home Page of the web application

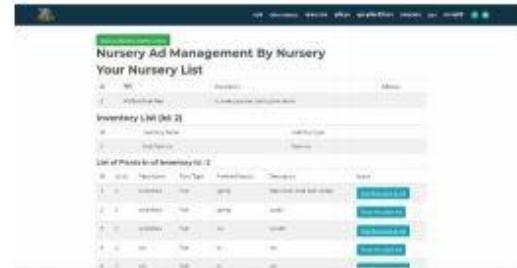


Fig. 4 Ad Management Portal for Nursery People

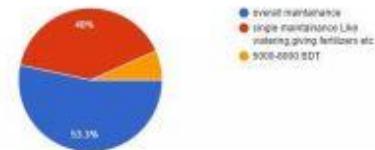
B. Survey Findings on the web and mobile Application

In our survey, most of the respondent was young people and almost 90% of the user are young people. We demonstrated our app to them and then the reviewed our app in terms of the features of the application. 50% of the respondents were aware of there should be service related to the diseases of the plants in the application. Almost 54% people wants overall maintenance of their garden.

4.Which service you want to get?
14 responses



6.If you are offered a service where service provider will do the daily maintenance(watering plants,spreading fertilizers and pesticides,overall maintenance) for your garden,how much would you like to pay monthly for that service?
15 responses



C. Errors in the application

Our system has several errors which we found during the deployment stage and UX testing phase. We observed problem during order tracking and order management. When a user orders a plant to buy the following 2 steps happen:

- Call the nursery caretaker to get the plant/other staff
- After that nursery owner update all their information on orders at days end.

This step creates problem sometimes so that user sometimes doesn't get update new information about the order.

CONCLUSION

Online platform for urban farming can be the potential application for creating green and beautiful Dhaka city. It can certainly help the end users with both economically and environmentally. Moreover, the software certainly created a

Page 4:

bridge between the customers and the nursery owners. Now also can the customers get new information about expert advices from home to grow their garden more effectively. It can reduce the overall cost of the maintains of their garden. Most interestingly, our app is economically good option for the both nursery owners and the customers. Since the nursery owners wants to promote their business, they can now use the online application and get profited. On the other hand, customers now can get home services via the online platform so that its also profitable for the home dwellers. In future, if we focus more on community building the software will have certain economic success.

ACKNOWLEDGMENT

This project is supervised by Mohammad Rezaul Islam. He was really supportive to us during the development of the project and we are really grateful to him for assisting us.

REFERENCES

- [1] Horst, Megan, Nathan McClintock, and Lesli Hoey. "The intersection of planning, urban agriculture, and food justice: a review of the literature." *Journal of the American Planning Association* 83.3 (2017): 277-295.
- [2] Lovell, Sarah Taylor. "Multifunctional urban agriculture for sustainable land use planning in the United States." *Sustainability* 2.8 (2010): 2499-2522.
- [3] Dos Santos, Maria José Palma Lampreia. "Smart cities and urban areas—Aquaponics as innovative urban agriculture." *Urban forestry & urban greening* 20 (2016): 402-406.
- [4] Bos, Elizabeth, and Luke Owen. "Virtual reconnection: The online spaces of alternative food networks in England." *Journal of rural studies* 45 (2016): 1-14.
- [5] Byomkesh, Talukder, Nobukazu Nakagoshi, and Ashraf M. Dewan. "Urbanization and green space dynamics in Greater Dhaka, Bangladesh." *Landscape and Ecological Engineering* 8.1 (2012): 45-58.
- [6] Lovell, Sarah Taylor, and John R. Taylor. "Supplying urban ecosystem services through multifunctional green infrastructure in the United States." *Landscape ecology* 28.8 (2013): 1447-1463.
- [7] Ansari, Mohammad Nayem Aziz. Opportunities and challenges of urban and peri-urban forestry and greening in Bangladesh: Dhaka city as a case. Diss. Sveriges lantbruksuniversitet, 2008.
- [8] Specht, Kathrin, et al. "Socially acceptable urban agriculture businesses." *Agronomy for sustainable development* 36.1 (2016): 17.

REFERENCES

APPENDIX A

- [1] Ansari, Mohammad Nayeem Aziz. "Opportunities and challenges of urban and peri-urban forestry and greening in Bangladesh: Dhaka city as a case ". *Diss. Sveriges lantbruksuniversitet*, 2008.
- [2] Byomkesh, Talukder, Nobukazu Nakagoshi, and Ashraf M. Dewan. "Urbanization and green space dynamics in Greater Dhaka, Bangladesh." *Landscape and Ecological Engineering* 8.1 (2012): 45-58.
- [3] Bos, Elizabeth, and Luke Owen. "Virtual reconnection: The online spaces of alternative food networks in England." *Journal of rural studies* 45 (2016): 1-14.
- [4] Dos Santos, Maria José Palma Lampreia. "Smart cities and urban areas—Aquaponics as innovative urban agriculture." *Urban forestry & urban greening* 20 (2016): 402-406.
- [5] Horst, Megan, Nathan McClintock, and Lesli Hoey. "The intersection of planning, urban agriculture, and food justice: a review of the literature." *Journal of the American Planning Association* 83.3 (2017): 277-295.
- [6] Kitaya, Yoshiaki, et al. "Rooftop farming with sweet potato for reducing urban heat island effects and producing food and fuel materials." *The Seventh International Conference on Urban Climate, Yokohama, Japan*. 2009.
- [6] Lovell, Sarah Taylor. "Multifunctional urban agriculture for sustainable land use planning in the United States." *Sustainability* 2.8 (2010): 2499-2522.
- [7] Lovell, Sarah Taylor, and John R. Taylor. "Supplying urban ecosystem services through multifunctional green infrastructure in the United States." *Landscape ecology* 28.8 (2013): 1447-1463.

- [8] Sanyé-Mengual, Esther, et al. "Rooftop farming: an opportunity towards urban sustainability." *Conference paper for ISIE2015: 8th biannual conference of the international society of industrial ecology. University of Surrey, Guilford, UK.* 2015.
- [9] Specht, Kathrin, et al. "Socially acceptable urban agriculture businesses." *Agronomy for sustainable development* 36.1 (2016): 17.
- [10] Yusoff, Norul Hafizah, Mohd Ramzi Mohd Hussain, and Izawati Tukiman. "Roles of community towards urban farming activities." *Planning Malaysia* 15.1 (2017).
- [11] Home, N. (n.d.). Home. Retrieved November 07, 2020, from <http://nagarkrishi.com/>
- [12] Homepage, PrakritiFarming. (n.d.). Home. Retrieved November 07, 2020, from <https://www.prakritifarming.com/>
- [13] Homepage, Ugaoo. (n.d.). Home. Retrieved November 07, 2020, from <https://www.ugaoo.com/>
- [14] Homepage, SamrtCultiva. (n.d.). Home. Retrieved November 07, 2020, from <https://www.smartcultiva.com/>
- [15] Homepage, SmartGardener. (n.d.). Home. Retrieved November 07, 2020, from <https://www.smartgardener.com/>

DATASHEET

APPENDIX B

Component Name	Details
Front End	It has blade template engine, JS files and CSS Files. Also views are structured in our convinent ways.
Back End	It contains the application layer with controller and model files
Data base	It's a mysql Database for the web application

Code structure:

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "WEB DEMO". Key files include config/filesystems.php, web.php, Kernel.php, and various controllers like blog.php, cultivation.php, customer.php, feature.php, item.php, login.php, material.php, nursery_catalog.php, nursery.php, NurseryInventoryModel.php, orders.php, Permission.php, plant_info.php, provider.php, resource.php, review.php, role.php, tracking.php, User.php, and bootstrap/config/database.php.
- Code Editor:** The active file is config/filesystems.php. The code defines storage configurations for local and AWS S3 drivers. It includes symbolic link definitions and environment variable placeholders.
- Status Bar:** A tooltip indicates "You have Windows Subsystem for Linux (WSL) installed on your system. Do you want to install the recommended extensions for it?" with "Install" and "Show Recommendations" buttons.

Code Samples:
Controllers:

```
<?php

namespace App\Http\Controllers;
```

```
use Illuminate\Support\Facades\Auth;
use App\User;
use App\customer;
use App\provider;
use App\orders;
use Illuminate\Http\Request;
use Session;
use View,Redirect,Validator,DB;

class HomeController extends Controller
{
    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function __construct()
    {
        // $this->middleware('auth');
    }

    /**
     * Show the application dashboard.
     *
     * @return \Illuminate\Contracts\Support\Renderable
     */
    public function index(){
        return view('pages.home',[ 'user'=>""]);
    }
    public function showUserHome()
    {
        $user = Auth::user();
        $id = Auth::id();
        Session::put('login_id', $id);
        Session::put('name', $user['name']);
        Session::put('phone_number', $user['phone_number']);
        Session::put('user_type', $user['user_type']);
    }
}
```

```

if($user['user_type']=="customer"){
    $name=explode(" ",$user['name']);
    if(sizeof($name)==3){
        $first_name=$name[0];
        $middle_name=$name[1];
        $last_name=$name[2];
    }
    else if(sizeof($name)==2){
        $first_name=$name[0];
        $last_name=$name[1];
    }
    else{
        $first_name=$name[0];
        $last_name="";
    }
    // var_dump($first_name[0]);
    /// exit();

    $new_customer= new customer;
    $new_customer->first_name= $first_name;
    $new_customer->last_name= $last_name;
    $new_customer->user_id= $id;
    $new_customer->phone_number= $user['phone_number'];
    $new_customer->address= "";
    $new_customer->bloger_id=$id;
    $new_customer->save();
    $new_customer->refresh();
    $customer_id= $new_customer->id;

Session::put('customer_id',$customer_id);
    return view('customer.customerHome',['user'=>$new_customer]);
}
elseif($user['user_type']=="provider"){

    $name=explode(" ",$user['name']);
    if(sizeof($name)==3){
        $first_name=$name[0];
        $middle_name=$name[1];
        $last_name=$name[2];
    }
    else if(sizeof($name)==2){
        $first_name=$name[0];
        $last_name=$name[1];
    }
}

```

```

        else{
            $first_name=$name[0];
            $last_name="";
        }

        $new_provider= new provider;
        $new_provider->first_name      = $first_name;
        $new_provider->last_name       = $last_name;
        $new_provider->user_id         = $id;
        $new_provider->phone_number   = $user['phone_number'];
        $new_provider->address        = "";
        $new_provider->is_broker=false;
        $new_provider->is_sourcer=false;
        $new_provider->bloger_id= $id;
        $new_provider->save();
        return view('providerModule.providerHome',[ 'user'=>$new_provider])
    ;
}

elseif($user['user_type']=="admin"){

    return view('admin.AdminLanding',[ 'user'=>$user]);
}
}

public function ShowListOfNurseries(){

    $nurseries_obj= DB::table('nurseries')->get();
    var_dump($nurseries_obj);
    exit();

}

public function ShowallPlantAd(){
    $mat_obj=DB::table('materials')->get();

    return View::make('customer.ShowAllPlantAd')-
>with('mat_obj',$mat_obj);

    var_dump($mat_obj);
    exit();

}

```

```

public function ShowOrderSummery(){

$order_obj=DB::table('orders')->where('id',Session::get('current_order_id'))->get();

return View::make('customer.ShowOrderSummery')->with('order_obj',$order_obj);

}

public function orderComplete(Request $request){

//placed', 'processing', 'delivered'
$mat_id= $request->input('mat_id');
$mat_obj=DB::table('materials')->where('id',$mat_id)->get();
$provider_obj=DB::table('providers')->where('id',$mat_obj[0]->owner_id)->get();
$provider_phone_number=$provider_obj[0]->phone_number;
$order_obj=new orders;
$order_obj->sender_id=Session::get('login_id');
$order_obj->recipient_id= $mat_obj[0]->owner_id;
$order_obj->tracking_id=$mat_obj[0]->owner_id.
Session::get('login_id').$mat_id;
$order_obj->order_summary=$mat_obj[0]->display_name .
' And The Amount is '. $request->input('quantity');
$unit_price=$mat_obj[0]->unit_price;
$qty=$unit_price*$request->input('quantity');
$order_obj->total_amount= $unit_price * $qty ;
$order_obj->order_status="placed";
$order_obj->provider_phone_number=$provider_phone_number;
$order_obj->save();
$order_obj->refresh();
Session::put('current_order_id',$order_obj->id);
Session::flash('message', 'Successfully Placed the Order!');
return Redirect::to('/ShowallPlantAd');
//var_dump($order_obj->total_amount);

}

public function orderItem($mat_id){

$mat_obj=DB::table('materials')->where('id',$mat_id)->get();
return View::make('customer.createOrderPage')->with('mat_obj',$mat_obj);
}

```

```

}

public function EditCustomer(Request $request){

    if($request->input('user_id')){

        $customer_info = DB::table('customers')->where('user_id',$request-
>input('user_id'))->get();

        $customer_id= $customer_info->id;
        $afftected= DB::table('customers')
            ->where('id', $customer_id)
            ->update(['first_name' => $request->input('first_name'),
            'last_name' => $request->input('last_name'),
            'user_id' => Session::get('login_id'),
            'address' => $request->input('address')]);
        // store the images
        //var_dump($afftected);

        $images=array();
        if($files=$request->file('images')){
            foreach($files as $file){
                $name=$file->getClientOriginalName();
                $url = Storage::url($name);
                $file_name = basename($name, ".jpg");
                // var_dump($file_name);
                //exit();
                $nursery_inventory_images_folder_name='Customer_img';
                $path = $file->store(
                    'Customer_img/'.Session::get('login_id')
                );

                // $file_show = Storage::get($path);
                // var_dump($file_show);
                // exit();
                $singleResource= new resource;
                $singleResource->img=$path;
                $singleResource->caption="The First name:". $request-
>input('first_name').
                                " and the Last name ". $request-
>input('last_name');
                $singleResource->customer_image_id=$customer_id;
                $singleResource->save();
            }
        }
    }
}

```

```

    }

    // exit();

}

return Redirect::to('/userLanding');

}

}

<?php

namespace App\Http\Controllers;
use Illuminate\Support\Facades\Auth;
use App\customer;
use App\resource;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Storage;
use View,Validator,Session,Redirect,DB;

class CustomerController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        //
        // get all the customers

        $user = Auth::user();
        $id = Auth::id();
        $customer_info = DB::table('customers')->where('user_id',$id)->get();

        // load the view and pass the nerds
        return View::make('customer.index')
            ->with('customers', $customer_info);
    }

    /**

```

```

 * Show the form for creating a new resource.
 *
 * @return \Illuminate\Http\Response
 */
public function create()
{
    //
    return View::make('customer.create');
}

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{

    $rules = array(
        'first_name'      => 'required',
        'last_name'       => 'required',
        'phone_number'    => 'required',
        'address'         => 'required|string'
    );
    $validator = Validator::make($request->all(), $rules);

    // process the login
    if ($validator->fails()) {
        return Redirect::to('customer/create')
            ->withErrors($validator)
            ->withInput(Input::except('name','email','password'));
    } else {
        // store
        $customer = new customer;
        $customer->first_name = $request->input('first_name');
        $customer->last_name  = $request->input('last_name');
        $customer->user_id    = Session::get('login_id');
        $customer->phone_number = $request->input('phone_number');
        $customer->address   = $request->input('address');
        $customer->bloger_id= Session::get('login_id');
        $customer->save();
        $customer->refresh();
    }
}

```

```

$customer_id_newly_created=$customer->id;

    // redirect
    Session::flash('message', 'Successfully created Customer!');
    return Redirect::to('customer');
}

}

/** 
 * Display the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function show(customer $customer)
{
    // show the view and pass the nerd to it
    return View::make('customer.show')
        ->with('customer', $customer);
}

/** 
 * Show the form for editing the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function edit(customer $customer)
{
    //

    //var_dump(Session::get('login_id'));

    return View::make('customer.edit')
        ->with('customer', $customer);
}

/** 
 * Update the specified resource in storage.

```

```

/*
 * @param \Illuminate\Http\Request $request
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function update(Request $request, customer $customer)
{
    //
}

/**
 * Remove the specified resource from storage.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function destroy(customer $customer)
{
    //
}
}

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

//use Validator,Redirect,Response;
use App\User;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Hash;

use Session;

class AuthController extends Controller
{
    //

    protected function username(){

        return "phone_number";
    }
}

```

```

}

public function showReg(){

    return view ("pages.regForm");
}

public function dologin(Request $request){

    // request()->validate([
    //     'name' => 'required'| 'string'| 'max:255',
    //     'phone_number' => 'required'|'string'|'phone_number'| 'max:255' | '
unique:users',
    //     'password' => 'required'|'string'| 'min:8'|'confirmed'
    //         ]);

    //     $credentials = $request->only('phone_number', 'password');
    //     if (Auth::attempt($credentials)) {
    //         // Authentication passed...
    //         return redirect()->intended('dashboard');
    //     }
    //     return Redirect::to("login")-
>withSuccess('Oppes! You have entered invalid credentials');

}

public function validation(Request $request){

    $data= $request->validate([
        'user_name' => 'required|string|max:255',
        'phone_number' => 'required |string | phone_number |max:255',
        'password' => 'required | string | min:6 |confirmed'

    ]);
    var_dump($data);

}

public function reg(Request $request){

    // return response()->json(['msg'=>'your data'],200);
}

```

```

var_dump( $request->all());
//exit();
$data= $request->validate([
    'user_name' => 'required|string|max:255',
    'phone_number' => 'required |string | phone_number |max:255',
    'password' => 'required | string | min:6 |confirmed'

]);
var_dump($data);
exit();

$this->validation($request);

return $request->all();

// request()->validate([
//     'name' => 'required'| 'string'| 'max:255',
//     'phone_number' => 'required'|'string'|'phone_number'| 'max:255'|'
unique:users',
//     'password' => 'required'||'string'| 'min:8'||'confirmed'
// ]);

//     $data = $request->all();

//     $check = $this->create($data);

//     return Redirect::to("home")-
>withSuccess('Great! You have Successfully loggedin');
}

public function home()
{
    if(Auth::check()){
        return view('home');
    }
    return Redirect::to("dologin")-
>withSuccess('Opps! You do not have access');
}

public function create(array $data)
{
    return User::create([
        'name' => $data['name'],

```

```

        'phone_number' => $data['phone_number'],
        'user_type'=>$data['user_type'],
        'password' => Hash::make($data['password']),
    ]);
}
}

<?php

namespace App\Http\Controllers;

use App\nursery_catalog;
use Illuminate\Http\Request;
use Session;

class Nursery_CatalogController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index($id)
    {
        //
        $catalog=DB::table('nursery_catalogs')->where('nursery_id',$id)->get();
        if($catalog){
            return View::make('providerModule.nursery.CatalogShow')
            ->with('catalog', $catalog);
        }
        else{
            return View::make('providerModule.nursery.CatalogShow')
            ->with('catalog', $catalog);
        }
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        //
    }
}

```

```
}

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{
    //
}

/**
 * Display the specified resource.
 *
 * @param \App\nursery_catalog $nursery_catalog
 * @return \Illuminate\Http\Response
 */
public function show(nursery_catalog $nursery_catalog)
{
    //
}

/**
 * Show the form for editing the specified resource.
 *
 * @param \App\nursery_catalog $nursery_catalog
 * @return \Illuminate\Http\Response
 */
public function edit(nursery_catalog $nursery_catalog)
{
    //
}

/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param \App\nursery_catalog $nursery_catalog
 * @return \Illuminate\Http\Response
 */
public function update(Request $request, nursery_catalog $nursery_catalog)
```

```

    {
        //
    }

    /**
     * Remove the specified resource from storage.
     *
     * @param \App\nursery_catalog $nursery_catalog
     * @return \Illuminate\Http\Response
     */
    public function destroy(nursery_catalog $nursery_catalog)
    {
        //
    }
}

<?php

namespace App\Http\Controllers;

use App\nursery;
use Illuminate\Http\Request;
use Session, Validator, View, Redirect, DB;
class NurseryController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        //
        $id=Session::get('login_id');
        $nurseries = DB::table('nurseries')->where('user_id', $id)->get();
        // $nurseries = nursery::find($id);

        if ($nurseries) {
            //
            return View::make('ProviderModule.nursery.showAllNurseryofUser') -
>with('nurseries', $nurseries);
        }
        else{

```

```

        return View::make('ProviderModule.nursery.createNursery');
    }

}

/**
 * Show the form for creating a new resource.
 *
 * @return \Illuminate\Http\Response
 */
public function create()
{
    //
    return View::make('ProviderModule.nursery.createNursery');

}

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{
    //
    $rules = array(
        'nursery_name'      => 'required',
        'description'       => 'required | string'

    );
    $validator = Validator::make($request->all(), $rules);

    // process the login
    if ($validator->fails()) {
        return Redirect::to('nursery_mng/create')
            ->withErrors($validator);
    } else {
        // store
        $nursery = new nursery;
        $nursery->nursery_name = $request->input('nursery_name');
        $nursery->description  = $request->input('description');
        $nursery->user_id      = Session::get('login_id');
    }
}

```

```

        $nursery->save();
        $nursery->refresh();
        Session::put('nursery_id', $nursery->id);
        //var_dump(Session::get('nursery_id'));
        // redirect
        Session::flash('message', 'Successfully created Nursery!');
        return Redirect::to('nursery_mng');
    }
}

/**
 * Display the specified resource.
 *
 * @param \App\nursery $nursery
 * @return \Illuminate\Http\Response
 */
public function show(nursery $nursery)
{
    //
    return View::make('ProviderModule.nursery.showSingleNursery')
        ->with('nursery', $nursery);
}

/**
 * Show the form for editing the specified resource.
 *
 * @param \App\nursery $nursery
 * @return \Illuminate\Http\Response
 */
public function edit(nursery $nursery)
{
    //

    return View::make('ProviderModule.nursery.editSingleNursery')
        ->with('nursery', $nursery);

}

/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param \App\nursery $nursery
 * @return \Illuminate\Http\Response
 */

```

```

        */
    public function update(Request $request, nursery $nursery)
    {
        //
    }

    /**
     * Remove the specified resource from storage.
     *
     * @param \App\nursery $nursery
     * @return \Illuminate\Http\Response
     */
    public function destroy(nursery $nursery)
    {
        //
    }
}

<?php

namespace App\Http\Controllers;

use App\NurseryInventoryModel;
use Illuminate\Http\Request;
use View,Session,Redirect,Validator,DB;

class NurseryInventoryController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        //

        $id=Session::get('nursery_id');
        var_dump($id);

        $inventories=DB::table('nursery_inventories')->where('nursery_id',$id)->get();
        // var_dump(count($inventories));
        // exit();
    }
}

```

```

if(count($inventories)>1){
    // var_dump("I am with value");
    // exit();
    $data=['inventories'=>$inventories,
    'nursery_id'=>$id];
    return View::make('providerModule.nursery.InventoryShowByNu
rseryId')
    ->with($data);
}
else{
    //var_dump("I am with no value");
    // exit();
    return View::make('providerModule.nursery.InventoryCreate')
    ->with('nursery_id', $id);
}
}

/**
 * Show the form for creating a new resource.
 *
 * @return \Illuminate\Http\Response
 */
public function create()
{
    //
    $id=Session::get('nursery_id');
    return View::make('providerModule.nursery.InventoryCreate')-
>with('nursery_id', $id);
}

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{
    //
    //
    var_dump(request()->all());
    // exit();
    $rules = array(

```

```

        'inv_name'      => 'required',
        'inv_type'       => 'required'

    );
$validator = Validator::make($request->all(), $rules);

// process the login
if ($validator->fails()) {
    return Redirect::to('nursery_inventory/create')

        ->withErrors($validator);
} else {
    //var_dump(request()->all());
    // exit();
    // store
    // var_dump($request->input('inv_name'));
    // exit();
    $nursery_inv = new NurseryInventoryModel;
    $nursery_inv->inv_name = $request->input('inv_name');
    $nursery_inv->inv_type = $request->input('inv_type');
    $nursery_inv->nursery_id = $request->input('nursery_id');

    $nursery_inv->save();
    $nursery_inv->refresh();

    Session::put('current_inv_id',$nursery_inv->id);

    // redirect
    Session::flash('message', 'Successfully created Nursery!');
    return Redirect::to('nursery_inventory');
}
}

/** 
 * Display the specified resource.
 *
 * @param \App\NurseryInventoryModel $nurseryInventoryModel
 * @return \Illuminate\Http\Response
 */
public function show($id)
{
    //
    //var_dump($id);
    $nurseryInventoryModel = NurseryInventoryModel::find($id);

```

```

    //var_dump($nurseryInventoryModel);
    $plant_infos=DB::table('plant_infos')-
>where('inv_id', $nurseryInventoryModel->id)->get();
    //var_dump($plant_infos);
    // exit();
    if($plant_infos){
        $data=['plant_infos'=>$plant_infos,
        'inventory'=>$nurseryInventoryModel];
        return View::make('ProviderModule.nursery.Inventory.showSingleNursery
Inventory')
            ->with($data);
    }
    else{
        return View::make('ProviderModule.nursery.Inventory.showSingleNursery
Inventory')
            ->with('inventory',$nurseryInventoryModel);
    }

}

/***
 * Show the form for editing the specified resource.
 *
 * @param \App\NurseryInventoryModel $nurseryInventoryModel
 * @return \Illuminate\Http\Response
 */
public function edit(NurseryInventoryModel $nurseryInventoryModel)
{
    //
    return View::make('ProviderModule.nursery.editSingleNursery')
        ->with('nursery', $nursery);
}

/***
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param \App\NurseryInventoryModel $nurseryInventoryModel
 * @return \Illuminate\Http\Response
 */
public function update(Request $request, NurseryInventoryModel $nurseryInvent
oryModel)
{
    //
}

```

```

    }

    /**
     * Remove the specified resource from storage.
     *
     * @param \App\NurseryInventoryModel $nurseryInventoryModel
     * @return \Illuminate\Http\Response
     */
    public function destroy(NurseryInventoryModel $nurseryInventoryModel)
    {
        //
    }
}

<?php

namespace App\Http\Controllers;

use App\plant_info;
use App\resource;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Storage;
use View,Session,Redirect,Validator,DB;

class Plant_InfosController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        //
        // $id=Session::get('nursery_id');
        $inv_id=Session::get('current_inv_id');
        $plant_infos=DB::table('plant_infos')->where('inv_id',$inv_id)->get();
        if($plant_infos){
            $data=['PlantList'=>$plant_infos,
                   'inv_id'=>$inv_id];
            return View::make('providerModule.nursery.Inventory.PlantList.PlantsS
howByInvId')
                ->with($data);
        }
    }
}

```

```

        else{

            return View::make('providerModule.nursery.Inventory.PlantList.PlantsC
reateByInvId')
                ->with($data);
            }
        }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        //
        $id=Session::get('current_inv_id');
        return View::make('providerModule.nursery.Inventory.PlantList.PlantsCreat
eByInvId')->with('inv_id', $id);
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        //
        $rules = array(
            'plant_name'      => 'required',
            'plant_type'      => 'required ',
            'plant_info'       => 'required '

        );
        $validator = Validator::make($request->all(), $rules);

        // process the login
        if ($validator->fails()) {
            return Redirect::to('nursery_inventory/create')
                ->withErrors($validator);
        } else {
    
```

```

$plant_info_by_inv = new plant_info;
$plant_info_by_inv->inv_id = $request->input('inv_id');
$plant_info_by_inv->plant_name = $request->input('plant_name');
$plant_info_by_inv->plant_type = $request->input('plant_type');
$plant_info_by_inv->prefered_season = $request-
>input('prefered_season');
$plant_info_by_inv->plant_info = $request->input('plant_info');
$plant_info_by_inv->save();
$plant_info_by_inv->refresh();
$plant_id_newly_created= $plant_info_by_inv->id;
// store the images
$images=array();

if($files=$request->file('images')){
    foreach($files as $file){
        $name=$file->getClientOriginalName();
        $url = Storage::url($name);
        $file_name = basename($name, ".jpg");
        var_dump($file_name);
        // exit();
        $nursery_inventory_images_folder_name='Nus_inv_img';
        $path = $file->store(
            'Nus_inv_img/'.$request-
>input('inv_id').'.'.$plant_id_newly_created
        );

        // $file_show = Storage::get($path);
        // var_dump($file_show);
        // exit();
        $singleResource= new resource;
        $singleResource->img=$path;
        $singleResource->caption="The Plant name:". $request-
>input('plant_name').
                                         " and the type is ". $request-
>input('plant_type');
        $singleResource->plant_image_id=$plant_id_newly_created;
        $singleResource->save();
    }
    // exit();
}
// redirect
Session::flash('message', 'Successfully Added to the inventory!');
return Redirect::to('plant_info');
}

```

```

    }

    /**
     * Display the specified resource.
     *
     * @param \App\plant_info $plant_info
     * @return Illuminate\Http\Response
     */
    public function show($id)
    {
        //
        $plant_info = plant_info::find($id);
        $inv_id=Session::get('current_inv_id');
        $nursery_id=Session::get('nursery_id');
        $provider_name=Session::get('name');
        $resource= DB::table('resources')->where('plant_image_id',$plant_info->id)->get();
        $num_of_images=$resource->count();
        // var_dump($nursery_id);
        // $path='public/Nus_inv_img/'. $inv_id.'/NoImage.bmp';
        // $NoPreviewimage=Storage::get($path);

        if($resource){
            //$file_show = Storage::get($resource->img);
            $data=[ 'plant_infos'=>$plant_info,
            'resources'=>$resource,
            'provider_name'=>$provider_name,
            'num_of_images'=>$num_of_images,
            'nursery_name'=>DB::table('nurseries')->where('id', $nursery_id)->first()];
            return View::make('providerModule.nursery.Inventory.PlantList.PlantShowById')
                ->with($data);
        }
        else{
            $data=[ 'plant_infos'=>$plant_info,
            'provider_name'=>$provider_name,
            'num_of_images'=>$num_of_images,
            'nursery_name'=>DB::table('nurseries')->where('id', $nursery_id)->first()];
            return View::make('providerModule.nursery.Inventory.PlantList.PlantShowById')
        }
    }
}

```

```

        ->with($data);
    }

}

/** 
 * Show the form for editing the specified resource.
 *
 * @param \App\plant_info $plant_info
 * @return \Illuminate\Http\Response
 */
public function edit(plant_info $plant_info)
{
    //
}

/** 
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param \App\plant_info $plant_info
 * @return \Illuminate\Http\Response
 */
public function update(Request $request, plant_info $plant_info)
{
    //
}

/** 
 * Remove the specified resource from storage.
 *
 * @param \App\plant_info $plant_info
 * @return \Illuminate\Http\Response
 */
public function destroy(plant_info $plant_info)
{
    //
}
}

<?php

namespace App\Http\Controllers;

```

```

use App\material;
use Illuminate\Http\Request;
use View,Session,Redirect,Validator,DB;

class ProviderServices extends Controller
{
    //

    public function __construct()
    {
        $this->middleware('auth');

    }

    public function createInvByNusId($id){
        $value= Session::get('nursery_id');
        if($value!=NULL){
            Session::forget('nursery_id');
        }

        Session::put('nursery_id',$id);
        if(count(DB::table('nursery_inventories')->where('nursery_id',$id)->get())>0){
            return Redirect::to('nursery_inventory');
        }
        return Redirect::to('nursery_inventory/create');

    }
    public function createPlantByInvId($inv_id){

        $value= Session::get('current_inv_id');

        if($value!=NULL){
            Session::forget('current_inv_id');
        }
        Session::put('current_inv_id',$inv_id);
        return Redirect::to('plant_info/create');

    }
    public function AdCreationHomePage(){
        var_dump("jsjdsjd");
        //exit();
        $user_id=Session::get('login_id');
    }
}

```

```

$user_name=Session::get('name');
var_dump($user_name);
$nursery_obj=DB::table('nurseries')->where('user_id',$user_id)->get();
var_dump(count($nursery_obj));
//exit();
if(count($nursery_obj)>0){
    var_dump("Loop 1");
    $data=['user_name'=>$user_id,
    'nursery_obj'=>$nursery_obj];

return View::make('ProviderModule.nursery.AdMng.Homepage')->with($data);

}
else{
    var_dump("loop 2");
    $data=['user_name'=>$user_id,
    'nursery_obj',null];
    //{$nursery_obj=null;
    return View::make('ProviderModule.nursery.AdMng.Homepage')-
>with($data);

}
}

public function showNurseryDetails($id){
// var_dump($id);
$nursery_obj=DB::table('nurseries')->where('id',$id)->get();
$inventory_obj=DB::table('nursery_inventories')->where('nursery_id',$id)-
>get();
$i=0;
$plant_info_obj=array();
foreach($inventory_obj as $key=>$item){

    $plant_info_obj[$i]=DB::table('plant_infos')->where('inv_id',$item-
>id)->get();

    if(count($plant_info_obj[$i])==0){
        //var_dump("Is 0");
        $is_inventory_empty[$item->id]=true;
    }
    else{
        $is_inventory_empty[$item->id]=false;
    }
    //var_dump($plant_info_obj[$key]);
    $i++;
}
}

```

```

    }

    $i=0;
    //exit();
    // foreach($plant_info_obj[0] as $item){
    //     var_dump( $plant_info_obj[$i][$item->id][0]->id);
    //     exit();
    // }

    $data= [
        'nursery_obj'=>$nursery_obj,
        'inv_obj'=>$inventory_obj,
        'plant_infos'=> $plant_info_obj,
        'is_inventory_empty'=>$is_inventory_empty];

    // var_dump($nursery_obj);
    return View::make('ProviderModule.nursery.AdMng.showNurseryInfo')-
>with($data);

    // var_dump( $plant_info_obj);

    //exit();

}

public function postAd($plant_id){

    // var_dump($plant_id);
    $owner_id=Session::get('login_id');
    $inv_obj=DB::table('plant_infos')->where('id',$plant_id)->get('inv_id');
    $inv_id= $inv_obj[0]->inv_id;

    $data=[ 'owner_id'=>$owner_id,'inv_id'=>$inv_id,'plant_id'=>$plant_id];

    return View::make('ProviderModule.nursery.AdMng.CreateAd')->with($data);

}

public function publishAd(Request $request){

    var_dump($request->all());
    $material=new material;
    $material->display_name= $request->input('display_name');
    $material->owner_id= $request->input('owner_id');
    $material->inv_id= $request->input('inv_id');
    $material->unit_price= $request->input('unit_price');
}

```

```

$material->display_info= $request->input('display_info');
$material->plant_id= $request->input('plant_id');
$material->save();
Session::flash('message', 'Successfully created Ad!');
$nursery_id=Session::get('nursery_id');

return Redirect::to('nurseryDetailsShow/'.$nursery_id);
}

public function showAd($plant_id){
    $mat_obj=DB::table('materials')->where('plant_id',$plant_id)->get();
    $plant_info_obj=DB::table('plant_infos')->where('id',$plant_id)->get();
    $resource= DB::table('resources')->where('plant_image_id',$plant_id)->get();
    $nursery_name=DB::table('nurseries')->where('id', Session::get('nursery_id'))->get('nursery_name');
    $num_of_images=$resource->count();
    if($num_of_images>0){
        //file_show = Storage::get($resource->img);
        $data=[

            'mat_obj'=>$mat_obj,
            'plant_info'=>$plant_info_obj,
            'resources'=>$resource,
            'nursery_id'=>Session::get('nursery_id'),
            'num_of_images'=>$num_of_images,
            'nursery_name'=>$nursery_name];
        return View::make('ProviderModule.nursery.AdMng.ShowAd')-
>with($data);
    }
    else{
        $data=[ 'plant_info'=>$plant_info_obj,
            'mat_obj'=>$mat_obj,
            'num_of_images'=>0,
            'nursery_id'=>Session::get('nursery_id'),
            'resources'=>$resource,
            'nursery_name'=>$nursery_name];
        return View::make('ProviderModule.nursery.AdMng.ShowAd')-
>with($data);
    }
}

}

public function showServices(){
    $user_id=Session::get('login_id');

```

```
        return View::make('ProviderModule.showListOfServices')-
>with('user_id',$user_id);

    }

}

<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use App\Providers\RouteServiceProvider;
use App\User;
use Illuminate\Foundation\Auth\RegistersUsers;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Validator;

class RegisterController extends Controller
{

    use RegistersUsers;

    /**
     * Where to redirect users after registration.
     *
     * @var string
     */
    protected $redirectTo = RouteServiceProvider::HOME;

    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function __construct()
    {
        $this->middleware('guest');
    }

    /**
     * Get a validator for an incoming registration request.
     *

```

```

 * @param array $data
 * @return \Illuminate\Contracts\Validation\Validator
 */
protected function validator(array $data)
{
    return Validator::make($data, [
        'name' => ['required', 'string', 'max:255'],
        'email' => ['string', 'email', 'max:255', 'unique:users'],
        'phone_number' => ['required', 'string', 'max:255', 'unique:users'],
        'password' => ['required', 'string', 'min:6', 'confirmed'],
    ], [
        'phone_number.required' => 'The phone_number is required.',
        'phone_number.unique' => 'The phone_number is registered in the system',
    ]);
}

/**
 * Create a new user instance after a valid registration.
 *
 * @param array $data
 * @return \App\User
 */
protected function create(array $data)
{
    return User::create([
        'name' => $data['name'],
        'email' => $data['email'],
        'phone_number' => $data['phone_number'],
        'user_type' => $data['user_type'],
        'password' => Hash::make($data['password']),
    ]);

    // $name=explode(" ",$data['name']);
    // var_dump($name);
    // $first_name=$name[0];
    // $middle_name=$name[1];
    // var_dump($first_name);
    // var_dump($middle_name);
    // exit();
}

```

```
    protected function redirectTo()
{
    return '/userLanding';
}

}
```

Migrations and DB connection tables:

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateNurseryInventoriesTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('nursery_inventories', function (Blueprint $table) {
            $table->increments('id');
            $table->integer('nursery_id');
            $table->string('inv_name');
            $table->enum('inv_type',
                ['Plant_inv', 'Material_inv','Soil_inv',
                'Pesticide_inv','Seed_inv']);
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
```

```

        Schema::dropIfExists('nursery_inventories');
    }
}

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateMaterialsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('materials', function (Blueprint $table) {

            $table->increments('id');
            $table->string('display_name');
            $table->integer('owner_id');
            $table->integer('unit_price');
            $table->longText('display_info');
            $table->date('mdf');
            $table->date('exd');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('materials');
    }
}

<?php

```

```
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateBlogsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('blogs', function (Blueprint $table) {

            $table->increments('id');
            $table->string('blog_title');
            $table->string('blog_introduction');
            $table->string('blog_conclusion');
            $table->string('blog_body');
            $table->integer('user_id');
            $table->integer('resource_id');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('blogs');
    }
}

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
```

```

class CreateOrdersTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('orders', function (Blueprint $table) {

            $table->increments('id');
            $table->integer('sender_id');
            $table->integer('recipient_id');
            $table->integer('tracking_id');
            $table->longText('order_summary');
            $table->double('total_amount',15,3);
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('orders');
    }
}

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateResourcesTable extends Migration
{
    /**
     * Run the migrations.
     *

```

```

        * @return void
        */
    public function up()
    {
        Schema::create('resources', function (Blueprint $table) {

            $table->increments('id');
            $table->char('img',100);
            $table->string('caption');
            $table->integer('customer_image_id')->nullable()->default(true);
            $table->integer('provider_image_id')->nullable()->default(true);
            $table->integer('plant_image_id')->nullable()->default(true);
            $table->integer('soil_image_id')->nullable()->default(true);
            $table->integer('seed_image_id')->nullable()->default(true);
            $table->integer('pesticide_image_id')->nullable()->default(true);
            $table->integer('material_image_id')->nullable()->default(true);
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('resources');
    }
}

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreatePlantInfosTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */

```

```
public function up()
{
    Schema::create('plant_infos', function (Blueprint $table) {

        $table->increments('id');
        $table->integer('inv_id');
        $table->string('plant_name');
        $table->string('plant_type');
        $table->string('prefered_season');
        $table->longText('plant_info');
        $table->timestamps();
    });
}

/**
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    Schema::dropIfExists('plant_infos');
}
```