A Project report on

AN ANDROID APPLICATION FOR ENHANCING AGRI-TOURISM AND WETLAND CONSERVATION THROUGH FARMER-CONSUMER ENGAGEMENT

Submitted in partial fulfillment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

By

P. GOUSIYA	**(204G1A0534)**
B. ANAND	**(204G1A0511)**
C. MANOJ REDDY	**(204G1A0554)**
K. BRAMHA TEJA	**(204G1A0524)**

Under the Guidance of

Mrs. G. Nagaleela **M.Tech**

Assistant Professor

Department of Computer Science & Engineering

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)
Rotarypuram Village, B K Samudram Mandal, Ananthapuramu - 515701

2023-2024

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY

(AUTONOMOUS)

(Affiliated to JNTUA, Accredited by NAAC with 'A' Grade, Approved by AICTE, New Delhi &
Accredited by NBA (EEE, ECE & CSE)
Rotarypuram Village, BK Samudram Mandal, Ananthapuramu-515701

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Certificate

This is to certify that the Project report entitled **AN ANDROID APPLICATION FOR ENHANCING AGRI-TOURISM AND WETLAND CONSERVATION THROUGH FARMER - CONSUMER ENGAGEMENT** is the bonafide work carried out by **P. Gousiya, B. Anand, C. Manoj Reddy, K. Bramha Teja** bearing Roll Number **204G1A0534, 204G1A0511, 204G1A0554, 204G1A0524** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** during the academic year 2023-2024.

Project Guide
Mrs. G. Nagaleela M.Tech.
Assistant Professor

Head of the Department
Mr. P. Veera Prakash M.Tech., (Ph.D)
Assistant Professor

Date:

External Examiner

Place: Rotarypuram

DECLARATION

We, Ms. P. Gousiya bearing reg no: 204G1A0534, Mr. B. Anand bearing reg no: 204G1A0511, Mr. C. Manoj Reddy bearing reg no: 204G1A0554, Mr. K. Bramha Teja bearing reg no: 204G1A0524 students of **SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY**, Rotarypuram, hereby declare that the dissertation entitled **"AN ANDROID APPLICATION FOR ENHANCING AGRI-TOURISM AND WETLAND CONSERVATION THROUGH FARMER-CONSUMER ENGAGEMENT"** embodies the report of our project work carried out by us during IV year Bachelor of Technology under the guidance of **Mrs. G. Nagaleela,** Assistant Professor, Department of CSE, and this work has been submitted for the partial fulfillment of the requirements for the award of the Bachelor of Technology degree.

The results embodied in this project have not been submitted to any other University of Institute for the award of any Degree or Diploma.

P. GOUSIYA	Reg no: 204G1A0534
B. ANAND	Reg no: 204G1A0511
C. MANOJ REDDY	Reg no: 204G1A0554
K. BRAMHA TEJA	Reg no: 204G1A0524

VISION & MISSION OF THE INSTITUTION

Vision:

To become a premier Educational Institution in India offering the best teaching and learning environment for our students that will enable them to become complete individuals with professional competency, human touch, ethical values, service motto, and a strong sense of responsibility towards environment and society at large.

Mission:

- Continually enhance the quality of physical infrastructure and human resources to evolvein to a center of excellence in engineering education.
- Provide comprehensive learning experiences that are conducive for the students to acquire professional competences, ethical values, life-long learning abilities and understanding of the technology, environment and society.
- Strengthen industry institute interactions to enable the students work on realistic problems and acquire the ability to face the ever-changing requirements of the industry.
- Continually enhance the quality of the relationship between students and faculty which is a key to the development of an exciting and rewarding learning environment in the college.

VISION & MISSION OF THE DEPARTMENT OF CSE

Vision:

To evolve as a leading department by offering best comprehensive teaching and learning practices for students to be self-competent technocrats with professional ethics and social responsibilities.

Mission:

DM 1: Continuous enhancement of the teaching-learning practices to gain profound knowledge in theoretical & practical aspects of computer science applications.

DM 2: Administer training on emerging technologies and motivate the students to inculcate self-learning abilities, ethical values and social consciousness to become competent professionals.

DM 3: Perpetual elevation of Industry-Institute interactions to facilitate the students to work on real-time problems to serve the needs of the society.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that we have now the opportunity to express our gratitude for all of them.

It is with immense pleasure that we would like to express our indebted gratitude to our Guide **Mrs. G. Nagaleela, Assistant Professor, Computer Science & Engineering**, who has guided us a lot and encouraged us in every step ofthe project work. We thank her for the stimulating guidance, constant encouragement and constructive criticism which have made possible to bring out this project work.

We express our deep felt gratitude to **Mr. C. Lakshminatha Reddy, Assistant Professor** and **Mr. M. Narasimhulu, Assistant Professor,** Project Coordinators for their valuable guidance and unstinting encouragement enabled us to accomplish our project successfully in time.

We are very much thankful to **Mr. P. Veera Prakash, Assistant Professor & Head of the Department, Computer Science & Engineering,** for his kind support and for providing necessary facilities to carry out the work.

We wish to convey our special thanks to **Dr. G. Bala Krishna, Principal** of **Srinivasa Ramanujan Institute of Technology** for giving the required information in doing our project work. Not to forget, We thank all other faculty and non- teaching staff, and our friends who had directly or indirectly helped and supported us in completing our project in time.

We also express our sincere thanks to the Management for providing excellent facilities**.**

Finally, we wish to convey our gratitude to our families who fostered all the requirements and facilities that we need.

Project Associates
204G1A0534
204G1A0511
204G1A0554
204G1A0524

ABSTRACT

To improve agricultural wetlands conservation and development, a wide range of positive measures should be undertaken. Sustainable tourism in agricultural wetlands, when properly managed and developed, plays a major part in protecting wetlands and supporting those working in and around wetlands through jobs and increasing incomes. The system will communicate with the farmers to visit their fields and the user will gain the knowledge in crop science and culture of the villages. Lack of capacity of receiving tourists, viewable product processing and valuable farm commodities, and well-educated practitioner are barriers to effective development of Agri-tourism in wetlands. In consideration of agricultural wetland characteristics, these barriers can be overcome with government rational plans, visit plants and take-away commodities, and professional practitioners' cultivation.

Keywords:

Agri-Tourism, Wetaland Conservation, Sustainable Tourism, Agricultural Commodities, Android App, Knowledge exchange, community Engagement.

CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

SDLC	Software Development Life Cycle
SDK	Software Development Kit
ADT	Android Development Tools
JDK	Java Development Kit
IDE	Integrated Development Environment
SRS	System Requirements Specification
DVM	Dalvik Virtual Machine
UML	Unified Modelling Language
UI	User Interface
AOSP	Android Open-Source Project

CHAPTER 1

INTRODUCTION

Wetlands are valuable because they clean the water, recharge water supplies, reduce flood risks, and provide fish and wildlife habitat. In addition, wetlands provide recreational opportunities (such as hunting, fishing, and boating), aesthetic benefits, sites for research and education, and commercial fishery benefits. With the rapid social development, tourism is widely recognized as among the world’s largest industries with strongest development momentum and largest scale in global economy. Tourism is one of wetland values, next to water regulation, flood control, or climate mitigation. And, wetland tourism supports those working in and around wetlands, through job and income creation. Importantly, fragile wetlands can be protected by sustainable tourism. Agri-tourism differentiates from conventional tourism.

Tourists enjoy practical participation in the process of food production, in the life of local residents and in a rural community. It satisfies the human cognitive need within farming production. When tourists saturate themselves in rural tourism, their emotional needs of novelty and curiosity are fully satisfied through knowing the way of cultivation and agricultural products and learning the indigenous knowledge of rural people. They have direct perception on flora and fauna, plant and animal products and the products’ processing. Moreover, Agri-tourist experience the idyllic countryside with the atmosphere of rusticity, silence and smells of farms and villages.

1.1 Problem Statement

In many regions, there exist a significant disconnect between urban consumers and rural farmers, resulting in a lack of understanding about agricultural practices, limited opportunities for knowledge exchange, and a reduced sense of community engagement.

Farmers are often struggle to find a platform to showcase their crops and connect with potential consumers. The system proposed tries overcome the drawbacks of existing systems. The solution we are proposing is to design an android application to communicate with the farmers. This system will communicate with the farmers to visit their fields and the user will gain the knowledge in crop science.

1.2 Objectives

To accomplish the project's purpose, the following particular objectives have been established.

- Creating a seamless and efficient Android App.
- Promoting agricultural awareness, knowledge exchange, and community engagement.

1.3 Scope of the Project

The following are the boundaries that have established in the proposed system which defines scope.

- Reduce the communication gap between urban consumers and rural farmers.
- The user will register and view the crop details then he can book a slot with farmers and view the status and location of that farmers.

CHAPTER 2

LITERATURE SURVEY

Agri-Tourism

[1] Karthik Dharamkar, Palve Gajanand

It can contribute to the overall income, cash flow and profitability of a farm by providing alternative income via farm products, and farming activities Presently, the urban population has been going for agri-tourism as a way of relief from the daily routine of big cities. For this reason, it is one of the most dynamic emerging markets in our country.

Farm Diversification Through Agri-tourism

[2] Nicole Vaugeois, Shannon Bence,Anna Romanova

This trend, if recognized by farmers, could be used to create new revenue for agriculture through agri-tourism. The purpose of this manual is to provide step by-step guidance for farmers who are interested in engaging in agritourism. It presents a number of tools and strategies to design and create memorable farm-based experiences that can attract and satisfy visitors. The guide also profiles numerous examples of agri-tourism ventures to encourage innovation within the industry.

Agro-tourism in Wetlands

[3] Yan Liu; Min Zhou

Lack of capacity of receiving tourists, viewable product processing and valuable farm commodities, and well-educated practitioner are barriers to effective development of agro-tourism in wetlands. In consideration of agricultural wetland characteristics, these barriers can be overcome with government rational plans, visit able plants and take-away commodities, and professional practitioners' cultivation.

Local environmental impact of wood combustion in agro-tourism structures

[4] Lucian Ionel Cioca; Ramona Giurea; Ioan Achim Moise; Ilaria Precazzini; Marco Ragazzi; Elena Cristina Rada 2017

A technological appliance that assures a good combustion process development together with a proper utilization and maintenance of the whole system assume significant role. A correct attention to the entire process reduces therefore the emissions' dangerousness and improves the energy efficiency in exploiting the fuel's heat value.

Agro – Tourism 2022

[5] Chandra Shekhar

Agrotourism is an economic strategy aimed at assisting villagers in establishing alternative sources of income and preserving their cultures. Agrotourism integrates agricultural and touristic activities. he benefits of agro-tourism are widely acknowledged by both the urban and rural communities. However, there are some problems in the way these centres are developed. In order to overcome these issues, the government should support the efforts of the farmers in developing such centres.

CHAPTER 3

METHODOLOGY

3.1 App Development Process:

The methodology for developing the Android application was comprehensive, encompassing user research, iterative design, and rigorous testing to ensure functionality and ease of use. User research began with a series of focus groups and surveys targeting both potential users and farmers to ascertain their needs, preferences, and technological capabilities. This initial research informed the design phase, which utilized the Agile software development framework, allowing for flexible and adaptive construction of the application with regular feedback loops. Prototyping was an integral part of the design process, with wireframes and mock-ups created to visualize the app's features. Subsequent testing methodologies included a combination of alpha and beta testing with both target user groups to refine the user interface (UI) and user experience (UX) design. Functional testing, usability testing, and performance testing were conducted to ensure the app's reliability, efficiency, and scalability.

3.2 Strategies for User-Farmer Interaction:

To facilitate seamless and efficient interaction between users and farmers, the app incorporates a real-time messaging system, allowing users to send queries and receive responses from farmers directly. A user-friendly interface prioritizes accessibility, ensuring that users of varying technological proficiency can navigate the app with ease. The app also includes a feature that allows users to schedule virtual farm visits, providing a live video feed from the farm, thus simulating a real agri-tourism experience. To further enhance this interaction, the app integrates a translation feature to overcome language barriers, thereby broadening the scope of user-farmer communication.

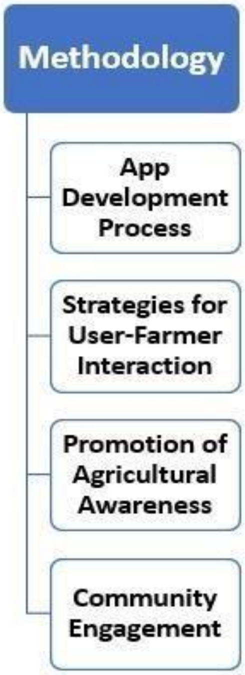

Methodology
App Development Process
Strategies for User-Farmer Interaction
Promotion of Agricultural Awareness
Community Engagement

Fig. 3.1: Methodology

3.3 Promotion of Agricultural Awareness and CommunityEngagement:

To promote agricultural awareness, the app features an educational portal with resources on crop science, sustainable farming practices, and the importance of wetland conservation. The content is curated to cater to a diverse audience, ranging from individuals with no prior agricultural knowledge to those seeking advanced understanding. Community engagement is stimulated through the app’s community forum, where users and farmers can discuss topics of mutual interest, share experiences, and organize community-driven conservation efforts. Gamification elements such as achievements and informational quests are incorporated to encourage users to engage with educational content and participate in community initiatives actively.

3.4 SOFTWARE DEVELOPMENT LIFE CYCLE

The meaning of Agile is swift or versatile. “Agile process model" refers to a software development approach based on iterative development. Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance. Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements. Each iteration involves a team working through a full software development life cycle including planning, requirements analysis, design, coding, and testing before a working product is demonstrated to the client.

Actually, Agile model refers to a group of development processes. These processes share some basic characteristics but do have certain subtle differences among themselves. A few Agile SDLC models are given below: Crystal A tern Feature-driven development Scrum Extreme programming (XP) Lean development Unified process In the Agile model, the requirements are decomposed into many small parts that can be incrementally developed.

The Agile model adopts Iterative development. Each incremental part is developed over an iteration. Each iteration is intended to be small and easily manageable and that can be completed within a couple of weeks only. At a time one iteration is planned, developed and deployed to the customers. Long-term plans are not made.

Agile model is the combination of iterative and incremental process models. Steps involve in agile SDLC models are:

- Requirement gathering
- Requirement Analysis
- Design Coding
- Unit testing
- Acceptance testing

The time to complete an iteration is known as a Time Box. Time-box refers to the maximum amount of time needed to deliver an iteration to customers. So, the end date for an iteration does not change. Though the development team can decide to reduce the delivered functionality during a Time-box if necessary to deliver it on time. The central principle of the Agile model is the delivery of an increment to the customer after each Time-box.

CHAPTER 4

SYSTEM REQUIREMENTS SPECIFICATIONS

4.1 H/W System Configuration: -

• Processor	-	I3/Intel Processor
• RAM	-	8 GB
• Hard Disk	-	1TB

S/W System Configuration: -

• Operating System	-	Windows 10
• JDK	-	java
• Plugin	-	Kotlin
• SDK	-	Android
• IDE	-	Android studio
• Server Script	-	PHP
• Database`	-	My SQL

4.2 Software Environment

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. Google Inc. purchased the initial developer of the software, Android Inc., in 2005.

Android's mobile operating system is based on the Linux kernel. Google and other members of the Open Handset Alliance collaborated on Android's development and release.

The Android Open-Source Project (AOSP) is tasked with the maintenance and further development of Android. The Android operating system is the world's best-selling Smartphone platform.

The Android SDK provides the tools and APIs necessary to begin developing applications Android platform using the Java programming language. Android has a large community of developers writing applications ("apps") that extend the functionality of the devices.

Features: -

- **Application framework** enabling reuse and replacement of components
- **Dalvik virtual machine** optimized for mobile devices
- **Integrated browser** based on the open-source Web Kit engine
- **Optimized graphics** powered by a custom 2D graphics library; 3D graphics based on the OpenGL ES 1.0 specification (hardware acceleration optional)
- **SQLite** for structured data storage
- **Media support** for common audio, video, and still image formats (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
- **GSM Telephony** (hardware dependent)
- **Bluetooth, EDGE, 3G, and WIFI** (hardware dependent)
- **Camera, GPS, compass, and accelerometer** (hardware dependent)
- **Rich development environment** including a device emulator, tools for debugging, memory and performance profiling, and a plugin for the Eclipse IDE

4.3 Android Architecture

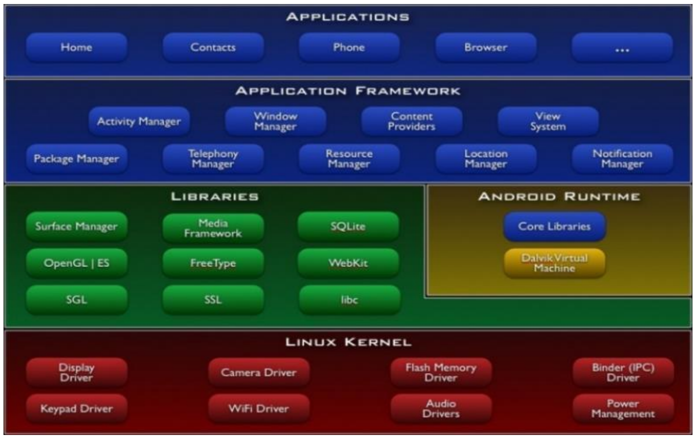


Fig. 4.1**:** Android Architecture

4.4 Libraries

Android includes a set of C/C++ libraries used by various components of the Android system. These capabilities are exposed to developers through the Android application framework. Some of the core libraries are listed below:

- **System C library** - a BSD-derived implementation of the standard C system library (libc), tuned for embedded Linux-based devices
- **Media Libraries** - based on Packet Video's Open CORE; the libraries support playback and recording of many popular audio and video formats, as well as static image files, including MPEG4, H.264, MP3, AAC, AMR, JPG, and PNG
- **Surface Manager** - manages access to the display subsystem and seamlessly composites 2D and 3D graphic layers from multiple applications
- **LibWebCore** - a modern web browser engine which powers both the Android browser and an embeddable web view
- **SGL** - the underlying 2D graphics engine
- **3D libraries** - an implementation based on OpenGL ES 1.0 APIs; the libraries use either hardware 3D acceleration (where available) or the included, highly optimized 3D software rasterizer
- **Free Type** - bitmap and vector font rendering
- **SQLite** - a powerful and lightweight relational database engine available to all applications

Android Runtime

Android includes a set of core libraries that provides most of the functionality available in the core libraries of the Java programming language.

Every Android application runs in its own process, with its own instance of the Dalvik virtual machine. Dalvik has been written so that a device can run multiple VMs efficiently. The Dalvik VM executes files in the Dalvik

Executable (.dex) format which is optimized for minimal memory footprint. The VM is register-based, and runs classes compiled by a Java language compiler that have been transformed into the. dex format by the included "dx" tool.

Linux Kernel

Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model. The kernel also acts as an abstraction layer between the hardware and the rest of the software stack.

The Linux kernel is an operating system kernel used by the Linux family of Unix-like operating systems. It is one of the most prominent examples of free and open-source software.

The Linux kernel is released under the GNU General Public License version 2 (GPLv2), (plus some firmware images with various licenses), and is developed by contributors worldwide. Day-to-day development takes place on the Linux kernel mailing list.

The Linux kernel was initially conceived and created by Finnish computer science student Linus Torvalds in 1991. Linux rapidly accumulated developers and users who adapted code from other free software projects for use with the new operating system. The Linux kernel has received contributions from thousands of programmers.[10] Many Linux distributions have been released based upon the Linux kernel.

The Linux kernel has extensive support for and runs on many virtual machine architectures both as the host operating system and as a guest operating system. The virtual machines usually emulate Intel x86 family of processors, though in a few cases PowerPC or ARM processors are also emulated.

At Google, the team led by Rubin developed a mobile device platform powered by the Linux kernel. Google marketed the platform to handset makers and carriers on the premise of providing a flexible, upgradable system. Google had lined up a series of hardware component and software partners and signaled to carriers that it was open to various degrees of cooperation on their part.[28][29][30]

Speculation about Google's intention to enter the mobile communications market continued to build through December 2006. Reports from the BBC and The Wall Street Journal noted that Google wanted its search and applications on mobile phones and it was working hard to deliver that. Print and online media outlets soon reported rumors that Google was developing a Google-branded handset.

Hardware running Android

The main supported platform for Android is the ARM architecture.

The Android OS can be used as an operating system for cellphones, netbooks and tablets, including the Dell Streak, Samsung Galaxy Tab, TV and other devices.[68][69] The first commercially available phone to run the Android operating system was the HTC Dream, released on 22 October 2008.[70] In early 2010 Google collaborated with HTC to launch its flagship [71] Android device, the Nexus One. This was followed later in 2010 with the Samsung-made Nexus S.

The early feedback on developing applications for the Android platform was mixed. Issues cited include bugs, lack of documentation, inadequate QA infrastructure, and no public issue-tracking system. (Google announced an issue tracker on 18 January 2008.) In December 2007, Merge Lab mobile startup founder Adam Macbeth stated, "Functionality is not there, is poorly documented or just doesn't work... It's clearly not ready for prime time." Despite this, Android-targeted applications began to appear the week after the platform was announced. The first publicly available application was the Snake game The Android Dev Phone is a SIM-unlocked and hardware-unlocked device that is designed for advanced developers. While developers can use regular consumer, devices purchased at retail to test and use their applications, some developers may choose not to use a retail device, preferring an unlocked or no-contract device.

The Android software development kit (SDK) includes a comprehensive set of development tools.[80] These include a debugger, libraries, a handset emulator (based on QEMU), documentation, sample code, and tutorials. The SDK is downloadable on the android developer website. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution), Mac OS X 10.4.9 or later, Windows XP or later. The officially supported integrated development environment (IDE) is Eclipse (currently 3.5 or 3.6) using the Android Development Tools (ADT) Plugin, though developers may use any text editor to edit Java and XML files then use command line tools (Java Development Kit and Apache Ant are required) to create, build and debug Android applications as well as control attached Android devices (e.g., triggering a reboot, installing software package(s) remotely).[81]

Android applications are packaged in .apk format and stored under /data/app folder on the Android OS (the folder is accessible to root user only for security reasons). APK package contains .dex files (compiled byte code files called Dalvik executables), resource files, etc.

4.5 Android Operation System

Android is an operating system based on Linux with a Java programming interface. It provides tools, e.g., a compiler, debugger and a device emulator as well as its own Java Virtual machine (Dalvik Virtual Machine - DVM). Android is created by the Open Handset Alliance which is led by Google.

Android uses a special virtual machine, e.g., the Dalvik Virtual Machine. Dalvik uses special bytecode. Therefore, you cannot run standard Java bytecode on Android. Android provides a tool "dx" which allows to convert Java Class files into "dex" (Dalvik Executable) files. Android applications are packed into an .apk (Android Package) file by the program "aapt" (Android Asset Packaging Tool) To simplify development Google provides the Android Development

Tools (ADT) for Eclipse. The ADT performs automatically the conversion from class to dex files and creates the apk during deployment.

Android supports 2-D and 3-D graphics using the OpenGL libraries and supports data storage in an SQLite database.

All Android applications run in its own process and under its own user id which is generated automatically by the Android system during deployment. Therefore, the application is isolated from other running applications and a misbehaving application cannot easily harm other Android applications.

Important Android components :

An Android application consists out of the following parts:

- Activity - Represents the presentation layer of an Android application, e.g., a screen which the user sees. An Android application can have several activities and it can be switched between them during runtime of the application.
- Views - The User interface of an Activities is built with widgets classes which inherent from "android. view. View". The layout of the views is managed by "android. view. View Groups".
- Services - perform background tasks without providing an UI. They can notify the user via the notification framework in Android.
- Content Provider - provides data to applications, via a content provider your application can share data with other applications. Android contains a SQLite DB which can serve as data provider
- Intents are asynchronous messages which allow the application to request functionality from other services or activities. An application can call directly a service or activity (explicit intent) or asked the Android system for registered services and applications for an intent (implicit intents). For example, the application could ask via an intent for a contact application. Application registers themself to an intent via an Intent

Filter. Intents are a powerful concept as they allow to create loosely coupled applications.

- Broadcast Receiver - receives system messages and implicit intents, can be used to react to changed conditions in the system. An application can register as a broadcast receiver for certain events and can be started if such an event occurs.
- A Java Virtual Machine (JVM) enables a set of computer software programs and data structures to use a virtual machine model for the execution of other computer programs and scripts. The model used by a JVM accepts a form of computer intermediate language commonly referred to as Java bytecode. This language conceptually represents the instruction set of a stack-oriented, capability architecture. Sun Microsystems states there are over 4.5 billion JVM-enabled devices
- A JVM can also execute bytecode compiled from programming languages other than Java. For example, Ada source code can be compiled to execute on a JVM. JVMs can also be released by other companies besides Oracle (the developer of Java) — JVMs using the "Java" trademark may be developed by other companies as long as they adhere to the JVM specification published by Oracle and to related contractual obligations.
- Java was conceived with the concept of WORA: "write once, run anywhere". This is done using the Java Virtual Machine. The JVM is the environment in which java programs execute. It is software that is implemented on non-virtual hardware and on standard operating systems.
- JVM is a crucial component of the Java platform, and because JVMs are available for many hardware and software platforms, Java can be both middleware and a platform in its own right, [clarification needed] hence the trademark write once, run anywhere. The use of the same bytecode for all platforms allows Java to be described as "compile once, run anywhere", as opposed to "write once, compile anywhere", which describes cross-platform compiled languages. A JVM also enables such features as automated exception handling, which provides "root-cause"

debugging information for every software error (exception), independent of the source code.

- A JVM is distributed along with a set of standard class libraries that implement the Java application programming interface (API). Appropriate APIs bundled together form the Java Runtime Environment (JRE).
- Java's execution environment is termed the Java Runtime Environment, or JRE.
- Programs intended to run on a JVM must be compiled into a standardized portable binary format, which typically comes in the form of .class files. A program may consist of many classes in different files. For easier distribution of large programs, multiple class files may be packaged together in a .jar file (short for Java archive).
- The Java application launcher, java, offers a standard way of executing Java code. Compare java.
- The JVM runtime executes .class or .jar files, emulating the JVM instruction set by interpreting it, or using a just-in-time compiler (JIT) such as Oracle's Hotspot. JIT compiling, not interpreting, is used in most JVMs today to achieve greater speed. There are also ahead-of-time compilers that enable developers to precompile class files into native code for particular platforms.
- Like most virtual machines, the Java Virtual Machine has a stack-based architecture akin to a microcontroller/microprocessor. However, the JVM also has low-level support for Java-like classes and methods, which amounts to a highly idiosyncratic memory model and capability-based architecture.

Download the Android SDK

Welcome Developers! If you are new to the Android SDK, please read the steps below, for an overview of how to set up the SDK.

If you're already using the Android SDK, you should update to the latest tools or platform using the *Android SDK and AVD Manager*, rather than downloading a new SDK starter package. See Adding SDK Components.

Here an overview of the steps you must follow to set up the Android SDK:

Install the SDK starter package from the table above. (If you're on Windows, download the installer for help with the initial setup.)

Install the ADT Plugin for Eclipse (if you'll be developing in Eclipse).

Add Android platforms and other components to your SDK.

Explore the contents of the Android SDK (optional).

To get started, download the appropriate package from the table above, then read the guide to Installing the SDK.

Installing the SDK

Step 1. Preparing Your Development Computer

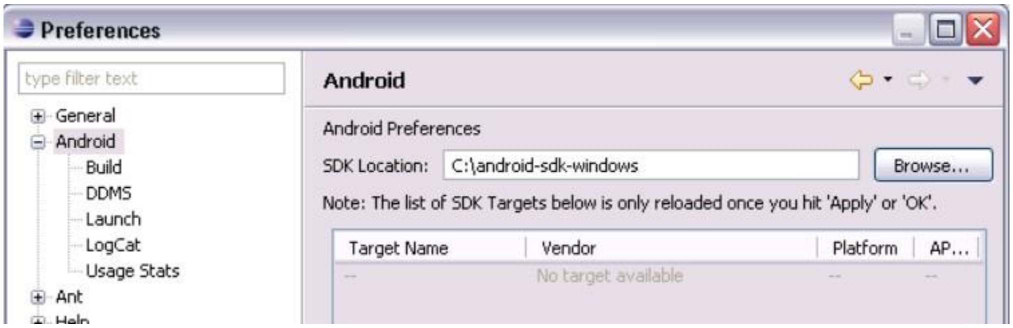
Before getting started with the Android SDK, take a moment to confirm that your development computer meets the System Requirements. In particular, you might need to install the JDK, if you don't have it already.

If you will be developing in Eclipse with the Android Development Tools (ADT) Plugin—the recommended path if you are new to Android—make sure that you have a suitable version of Eclipse installed on your computer as described in the System Requirements document. If you need to install Eclipse, you can download it from this location:

The "Eclipse Classic" version is recommended. Otherwise, a Java or RCP version of Eclipse is recommended.

Use the Eclipse update manager to install all available plugins for the Android Development Tools (ADT) from the URL https://dl-ssl.google.com/android/eclipse/. Configuration.

In Eclipse open the Preferences dialog via Windows -> Preferences. Select Android and maintain the installation path of the Android SDK.



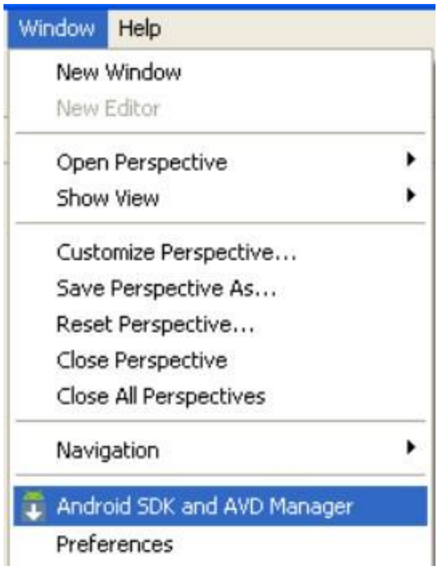
Android Preferences

SDK Location: C:\android-sdk-windows

Note: The list of SDK Targets below is only reloaded once you hit 'Apply' or 'OK'.

Target Name	Vendor	Platform	AP...
--	No target available	--	--

Select Window -> Android SDK and AVD Manager from the menu.



Select available packages and select the latest version of the SDK.

Step 2. Downloading the SDK Starter Package

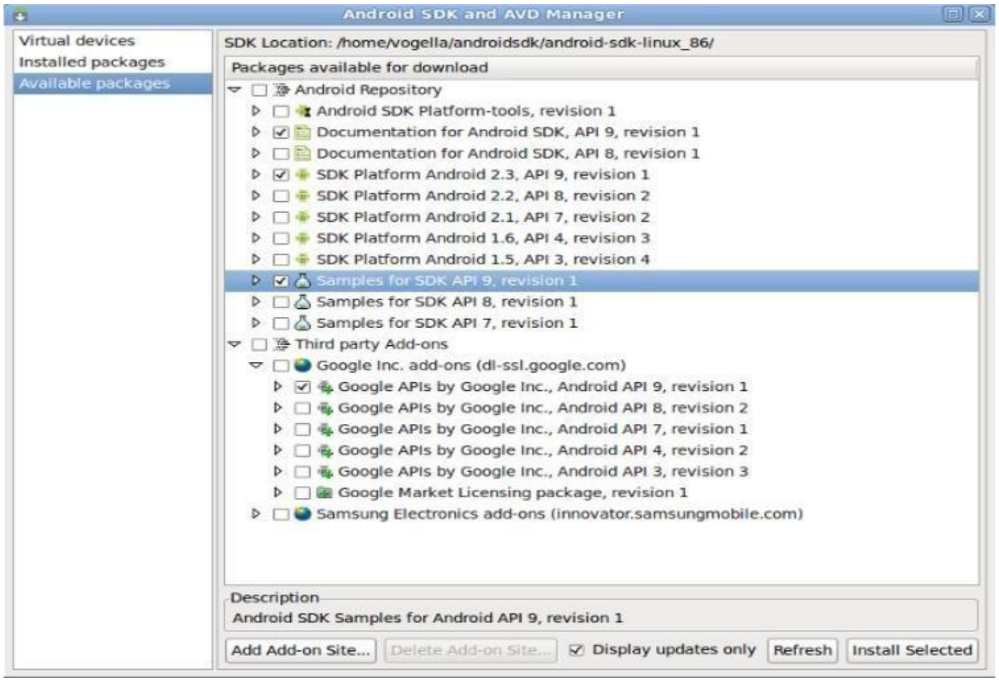
The SDK starter package is not a full development environment—it includes only the core SDK Tools, which you can use to download the rest of the SDK components (such as the latest Android platform).

If you haven't already, get the latest version of the SDK starter package from the SDK download page.

If you downloaded a `.zip` or `.tgz` package (instead of the SDK installer), unpack it to a safe location on your machine. By default, the SDK files are unpacked into a directory named `android-sdk-<machine-platform>`.

If you downloaded the Windows installer (`.exe` file), run it now and it will check whether the proper Java SE Development Kit (JDK) is installed (installing it, if necessary), then install the SDK Tools into a default location (which you can modify).

Make a note of the name and location of the SDK directory on your system—you will need to refer to the SDK directory later, when setting up the ADT plugin and when using the SDK tools from the command line.



SDK Location: /home/vogella/androidsdk/android-sdk-linux_86/

Packages available for download

- □ Android Repository
  - □ Android SDK Platform-tools, revision 1
  - ☑ Documentation for Android SDK, API 9, revision 1
  - □ Documentation for Android SDK, API 8, revision 1
  - ☑ SDK Platform Android 2.3, API 9, revision 1
  - □ SDK Platform Android 2.2, API 8, revision 2
  - □ SDK Platform Android 2.1, API 7, revision 2
  - □ SDK Platform Android 1.6, API 4, revision 3
  - □ SDK Platform Android 1.5, API 3, revision 4
  - ☑ Samples for SDK API 9, revision 1
  - □ Samples for SDK API 8, revision 1
  - □ Samples for SDK API 7, revision 1
- □ Third party Add-ons
  - □ Google Inc. add-ons (dl-ssl.google.com)
    - ☑ Google APIs by Google Inc., Android API 9, revision 1
    - □ Google APIs by Google Inc., Android API 8, revision 2
    - □ Google APIs by Google Inc., Android API 7, revision 1
    - □ Google APIs by Google Inc., Android API 4, revision 2
    - □ Google APIs by Google Inc., Android API 3, revision 3
    - □ Google Market Licensing package, revision 1
  - □ Samsung Electronics add-ons (innovator.samsungmobile.com)

Description
Android SDK Samples for Android API 9, revision 1

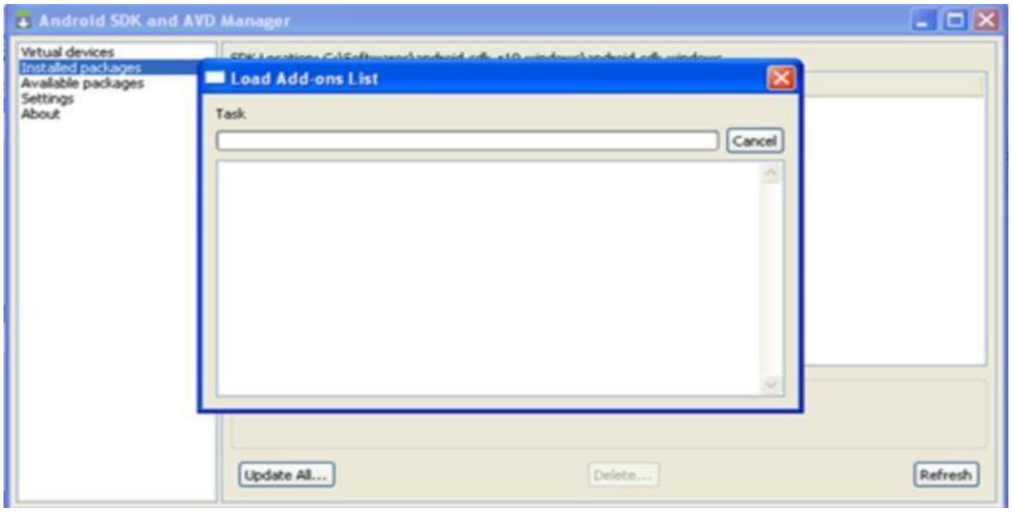
☑ Display updates only

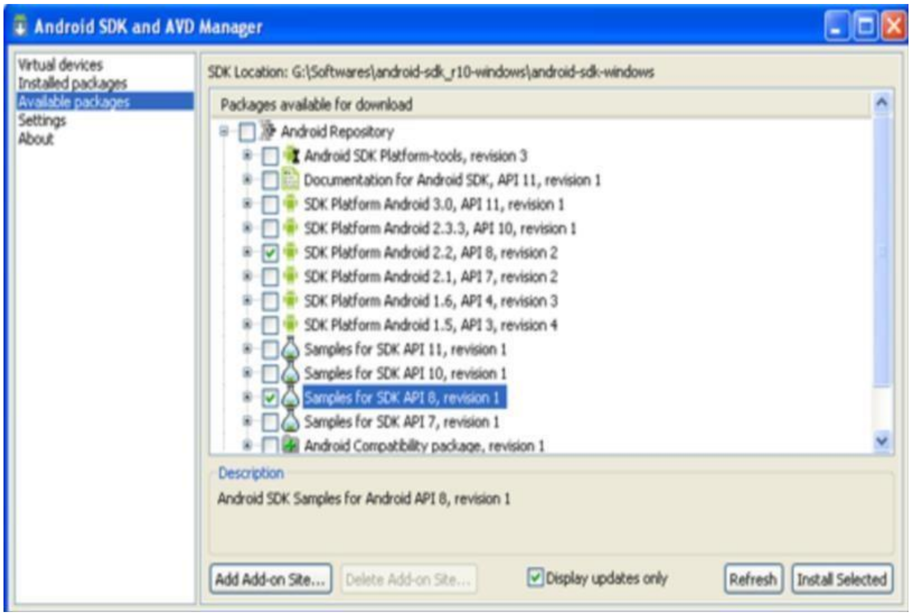
Step 3. Installing the ADT Plugin for Eclipse

Android offers a custom plugin for the Eclipse IDE, called Android Development Tools (ADT), that is designed to give you a powerful, integrated environment in which to build Android applications. It extends the capabilities of Eclipse to let you quickly set up new Android projects, create an application UI, debug your applications using the Android SDK tools, and even export signed (or unsigned) APKs in order to distribute your application. In general, developing in Eclipse with ADT is a highly recommended approach and is the fastest way to get started with Android.

If you'd like to use ADT for developing Android applications, install it now. Read Installing the ADT Plugin for step-by-step installation instructions, then return here to continue the last step in setting up your Android SDK.

If you prefer to work in a different IDE, you do not need to install Eclipse or ADT. Instead, you can directly use the SDK tools to build and debug your application. The Introduction to Android application development outlines the major steps that you need to complete when developing in Eclipse or other IDEs.





SDK Location: G:\Softwares\android-sdk_r10-windows\android-sdk-windows

Packages available for download

- □ Android Repository
  - □ Android SDK Platform-tools, revision 3
  - □ Documentation for Android SDK, API 11, revision 1
  - □ SDK Platform Android 3.0, API 11, revision 1
  - □ SDK Platform Android 2.3.3, API 10, revision 1
  - ☑ SDK Platform Android 2.2, API 8, revision 2
  - □ SDK Platform Android 2.1, API 7, revision 2
  - □ SDK Platform Android 1.6, API 4, revision 3
  - □ SDK Platform Android 1.5, API 3, revision 4
  - □ Samples for SDK API 11, revision 1
  - □ Samples for SDK API 10, revision 1
  - ☑ Samples for SDK API 8, revision 1
  - □ Samples for SDK API 7, revision 1
  - □ Android Compatibility package, revision 1

Description

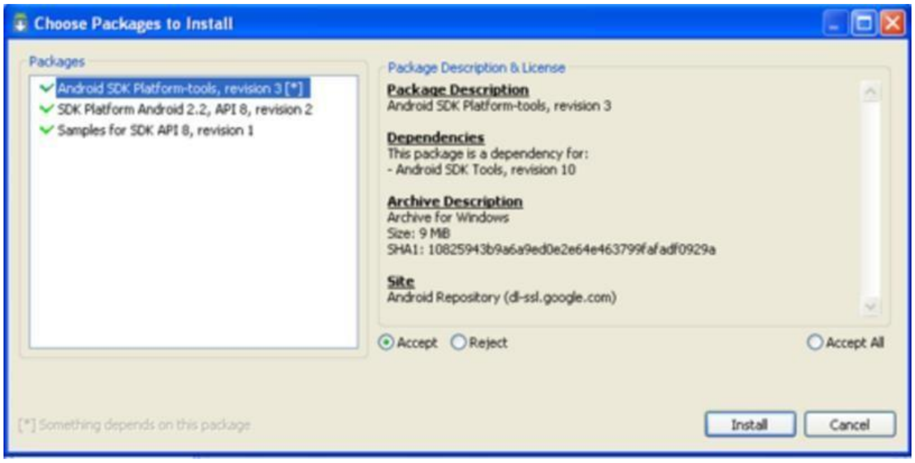
Android SDK Samples for Android API 8, revision 1

☑ Display updates only

Step 4. Adding Platforms and Other Components

The last step in setting up your SDK is using the Android SDK and AVD Manager (a tool included in the SDK starter package) to download essential SDK components into your development environment.

The SDK uses a modular structure that separates the major parts of the SDK Android platform versions, add-ons, tools, samples, and documentation into a set of separately installable components. The SDK starter package, which you've already downloaded, includes only a single component: the latest version of the SDK Tools. To develop an Android application, you also need to download at least one Android platform and the associated platform tools. You can add other components and platforms as well, which is highly recommended.



Choose Packages to Install

Packages

- ✓ Android SDK Platform-tools, revision 3 [*]
- ✓ SDK Platform Android 2.2, API 8, revision 2
- ✓ Samples for SDK API 8, revision 1

Package Description & License

Package Description
Android SDK Platform-tools, revision 3

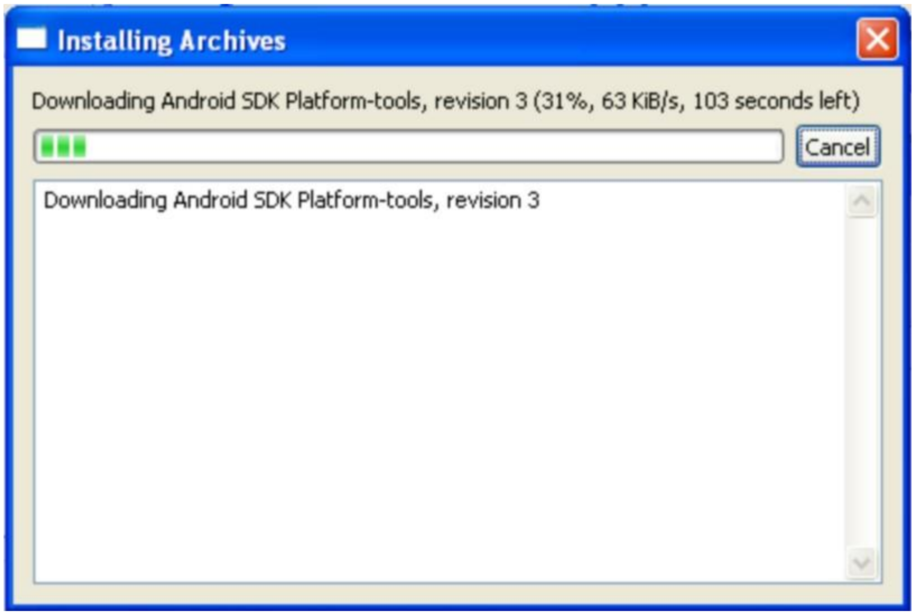
Dependencies
This package is a dependency for:
- Android SDK Tools, revision 10

Archive Description
Archive for Windows
Size: 9 MiB
SHA1: 10825943b9a6a9ed0e2e64e463799fafadf0929a

Site
Android Repository (dl-ssl.google.com)

● Accept ○ Reject ○ Accept All

[*] Something depends on this package



Installing Archives

Downloading Android SDK Platform-tools, revision 3 (31%, 63 KiB/s, 103 seconds left)

Downloading Android SDK Platform-tools, revision 3

If you used the Windows installer, when you complete the installation wizard, it will launch the Android SDK and AVD Manager with a default set of platforms and other components selected for you to install. Simply click **Install** to accept the recommended set of components and install them. You can then skip to Step 5, but we recommend you first read the section about the Available Components to better understand the components available from the Android SDK and AVD Manager.

You can launch the Android SDK and AVD Manager in one of the following ways:

- From within Eclipse, select **Window > Android SDK and AVD Manager**.
- On Windows, double-click the `SDK Manager.exe` file at the root of the Android SDK directory.
- On Mac or Linux, open a terminal and navigate to the `tools/` directory in the Android SDK, then execute: To download components, use the graphical UI of the Android SDK and AVD Manager to browse the SDK repository and select new or updated components (see figure 1). The Android SDK and AVD Manager installs the selected components in your SDK environment. For information about which components you should download, see Recommended Components.

The *Android Repository* offers these types of components:

- **SDK Tools** — Contains tools for debugging and testing your application and other utility tools. These tools are installed with the Android SDK starter package and receive periodic updates. You can access these tools in the <sdk>/tools/ directory of your SDK. To learn more about them, see SDK Tools in the developer guide.
- **SDK Platform-tools** — Contains platform-dependent tools for developing and debugging your application. These tools support the latest features of the Android and platform are typically updated only when a new platform becomes available. You can access these tools in the <sdk>/platform-tools/ directory. To learn more about them, see Platform Tools in the developer guide.
- **Android platforms** — An SDK platform is available for every production Android platform deployable to Android-powered devices. Each SDK platform component includes a fully compliant Android library, system image, sample code, and emulator skins. To learn more about a specific platform, see the list of platforms that appears under the section "Downloadable SDK Components" on the left part of this page.

- **USB Driver for Windows** (Windows only) — Contains driver files that you can install on your windows computer, so that you can run and debug your applications on an actual device. You *do not* need the USB driver unless you plan to debug your application on an actual Android-powered device. If you develop on Mac OS X or Linux, you do not need a special driver to debug your application on an Android-powered device. See Using Hardware Devices for more information about developing on a real device.
- **Samples** — Contains the sample code and apps available for each Android development platform. If you are just getting started with Android development, make sure to download the samples to your SDK.
- **Documentation** — Contains a local copy of the latest multi-version documentation for the Android framework API.

The *Third-party Add-ons* provide components that allow you to create a development environment using a specific Android external library (such as the Google Maps library) or a customized (but fully compliant) Android system image. You can add additional Add-on repositories by clicking **Add Add-on Site**.

ECLIPSE:

Eclipse is an open-source community whose projects are focused on building an extensible development platform, runtimes and application frameworks for building, deploying and managing software across the entire software lifecycle. Many people know us, and hopefully love us, as a Java IDE but Eclipse is much more than a Java IDE.

The Eclipse open-source community has over 60 open-source projects. These projects can be conceptually organized into seven different "pillars" or categories:

1. Enterprise Development
2. Embedded and Device Development
3. Rich Client Platform
4. Rich Internet Applications
5. Application Frameworks
6. Application Lifecycle Management (ALM)
7. Service Oriented Architecture

The Eclipse community is also supported by a large and vibrant ecosystem of major IT solution providers, innovative start-ups, universities and research institutions and individuals that extend, support and complement the Eclipse Platform.

The exciting thing about Eclipse is many people are using Eclipse in ways that we have never imagined. The common thread is that they are building innovative, industrial strength software and want to use great tools, frameworks and runtimes to make their job easier.

Eclipse is a multi-language software development environment comprising an integrated development environment (IDE) and an extensible plug-in system. It is written mostly in Java and can be used to develop applications in Java and, by means of various plug-ins, other programming languages including Ada, C, C++, COBOL, Perl, PHP, Python, Ruby (including Ruby on Rails framework), Scala, Clojure, and Scheme. The IDE is often called Eclipse ADT for Ada, Eclipse CDT for C/C++, Eclipse JDT for Java, and Eclipse PDT for PHP.

ARCHITECTURE:

Eclipse employs plug-ins in order to provide all of its functionality on top of (and including) the runtime system, in contrast to some other applications where functionality is typically hard coded. The runtime system of Eclipse is based on Equinox, an OSGi standard compliant implementation.

This plug-in mechanism is a lightweight software componentry framework. In addition to allowing Eclipse to be extended using other programming languages such as C and Python, the plug-in framework allows Eclipse to work with typesetting languages like LaTeX,[2] networking applications such as telnet, and database management systems. The plug-in architecture supports writing any desired extension to the environment, such as for configuration management. Java and CVS support is provided in the Eclipse SDK, with Subversion support provided by third-party plug-ins.

With the exception of a small run-time kernel, everything in Eclipse is a plug-in. This means that every plug-in developed integrates with Eclipse in exactly the same way as other plug-ins; in this respect, all features are "created equal". Eclipse provides plug-ins for a wide variety of features, some of which are through third parties using both free and commercial models. Examples of plug-ins include a UML plug-in for Sequence and other UML diagrams, a plug-in for DB Explorer, and many others.

The Eclipse SDK includes the Eclipse Java Development Tools (JDT), offering an IDE with a built-in incremental Java compiler and a full model of the Java source files. This allows for advanced refactoring techniques and code analysis. The IDE also makes use of a workspace, in this case a set of metadata over a flat file space allowing external file modifications as long as the corresponding workspace "resource" is refreshed afterwards.

Eclipse implements widgets through a widget toolkit for Java called SWT, unlike most Java applications, which use the Java standard Abstract Window Toolkit (AWT) or Swing. Eclipse's user interface also uses an intermediate GUI layer called JFace, which simplifies the construction of applications based on SWT.

Rich Client Platform

- Equinox OSGi – a standard bundling framework.
- Core platform – boot Eclipse, run plug-ins
- Standard Widget Toolkit (SWT) – a portable widget toolkit
- JFace – viewer classes to bring model view controller programming to SWT, file buffers, text handling, text editors
- Eclipse Workbench – views, editors, perspectives, wizards history.
- Eclipse began as an IBM Canada project. It was developed by Object Technology International (OTI) as a Java-based replacement for the Smalltalk based Visual Age family of IDE products, which itself had been developed by OTI. In November 2001, a consortium was formed to further the development of Eclipse as open source. In January 2004, the Eclipse Foundation was created.

Eclipse 3.0 (released on 21 June 2004) selected the OSGi Service Platform specifications as the runtime architecture.

Eclipse was originally released under the Common Public License, but was later relicensed under the Eclipse Public License. The Free Software Foundation has said that both licenses are free software licenses, but are incompatible with the GNU General Public License (GPL). Mike Milinkovich, of the Eclipse Foundation commented that moving to the GPL would be considered when version 3 of the GPL was released.

According to Lee Nackman, Chief Technology Officer of IBM's Rational division at that time and later head of Rational software development and support, the name "Eclipse" was chosen to target Microsoft's Visual Studio product, and not Sun Microsystems. Ironically, Nackman is now himself a Microsoft employee.

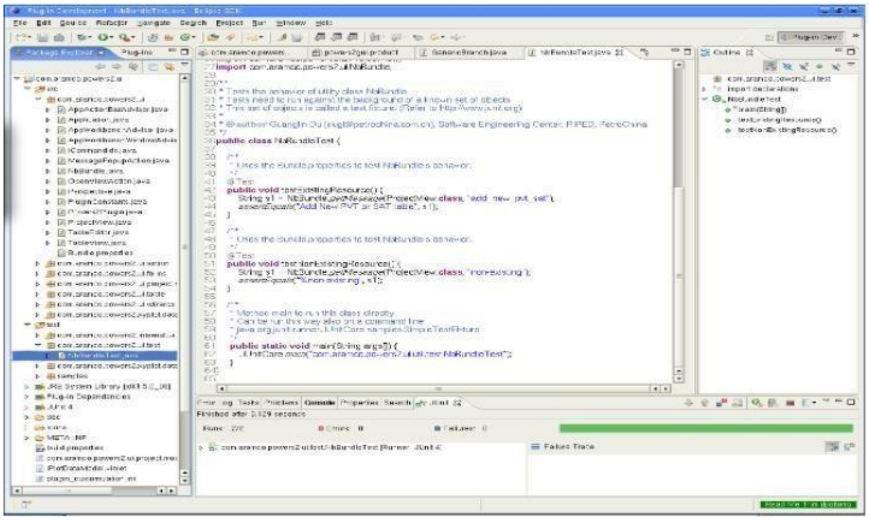
Eclipse (SDK)

Eclipse Software Development Kit (SDK) is a Java based open-source integrated development environment (IDE) which combines a number of different Eclipse projects including Platform, Java Development Tools (JDT) and the Plug-in Development Environment (PDE).

Eclipse can be used to create a large array of software applications using languages ranging from PHP, C++ programs, to Java. It is one of the most popular development tools in both the open-source and commercial worlds.

It provides Java editing with validation, incremental compilation, cross-referencing, code assist; an XML Editor; Mylyn; and much more.

Eclipse is released under the Eclipse Foundation, a commercially friendly license that allows organizations to include Eclipse software in their commercial products, while at the same time asking those who create derivative works of EPL code to contribute back to the community.



```
import com.arance.powers2.ui.NbBundle;

/**
 * Tests the behavior of utility class NbBundle.
 * Tests need to run against the background of a known set of objects.
 * This set of objects is called a test fixture. (Refer to http://www.junit.org)
 *
 * @author GuangIn Du ([illegible]@petrochina.com.cn), Software Engineering Center, RIPED, PetroChina
 */
public class NbBundleTest {

    /**
     * Uses the Bundle.properties to test NbBundle's behavior.
     */
    @Test
    public void testExistingResource() {
        String s1 = NbBundle.getMessage(ProjectView.class, "add_new_pvt_sat");
        assertEquals("Add New PVT or SAT table", s1);
    }

    /**
     * Uses the Bundle.properties to test NbBundle's behavior.
     */
    @Test
    public void testNonExistingResource() {
        String s1 = NbBundle.getMessage(ProjectView.class, "nonexisting");
        assertEquals("!nonexisting", s1);
    }

    /**
     * Method main to run this class directly.
     * Can be run this way also on a command line:
     * java org.junit.runner.JUnitCore samples.SimpleTestFixture
     */
    public static void main(String args[]) {
        JUnitCore.main("com.arance.powers2.ui.test.NbBundleTest");
    }
```

```
Finished after 2.129 seconds
Runs: 2/2        Errors: 0        Failures: 0
com.arance.powers2.ui.test.NbBundleTest [Runner: JUnit 4]
```

Plug-in Development - NbBundleTest.java - Eclipse SDK
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Plug-ins
Outline
Problems
Console
Properties
Search
JUnit
Failure Trace

Fig. 4.2: Eclipse Platform

Eclipse Platform

The Eclipse Platform provides the core frameworks and services upon which all plug-in extensions are created. It also provides the runtime in which plug-ins are loaded, integrated, and executed. The primary purpose of the Platform is to enable other tool developers to easily build and deliver integrated tools.

Features include:

- Supports the construction of a variety of tools for application development
- Supports an unrestricted set of tool providers, including independent software vendors (ISVs)
- Supports tools to manipulate arbitrary content types (e.g., HTML, Java, C, JSP, EJB, XML, and GIF)
- Facilitates seamless integration of tools within and across different content types and tool providers
- Supports both GUI and non-GUI-based application development environments.

Java Development Tools (JDT)

The JDT project provides the tool plug-ins that implement a Java IDE supporting the development of any Java application, including Eclipse plug-ins. It adds a Java project nature and Java perspective to the Eclipse Workbench as well as a number of views, editors, wizards, builders, and code merging and refactoring tools. The JDT project allows Eclipse to be a development environment for itself.

Features include:

- Java projects with source files arranged in package directories
- Editing with keyword and syntax colouring, outline showing declaration structure
- Code formatter
- Refactoring

- Search
- Compare
- Compile - JCK-compliant Java compiler
- Run Java programs in a separate target Java virtual machine
- Debug programs with JPDA-compliant Java virtual machine

Android Source Code

The following step is optional.

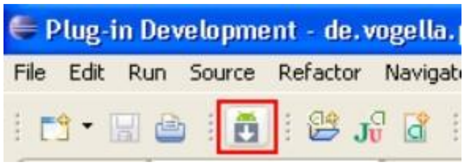
During Android development it is very useful to have the Android source code available as Android uses a lot of defaults.

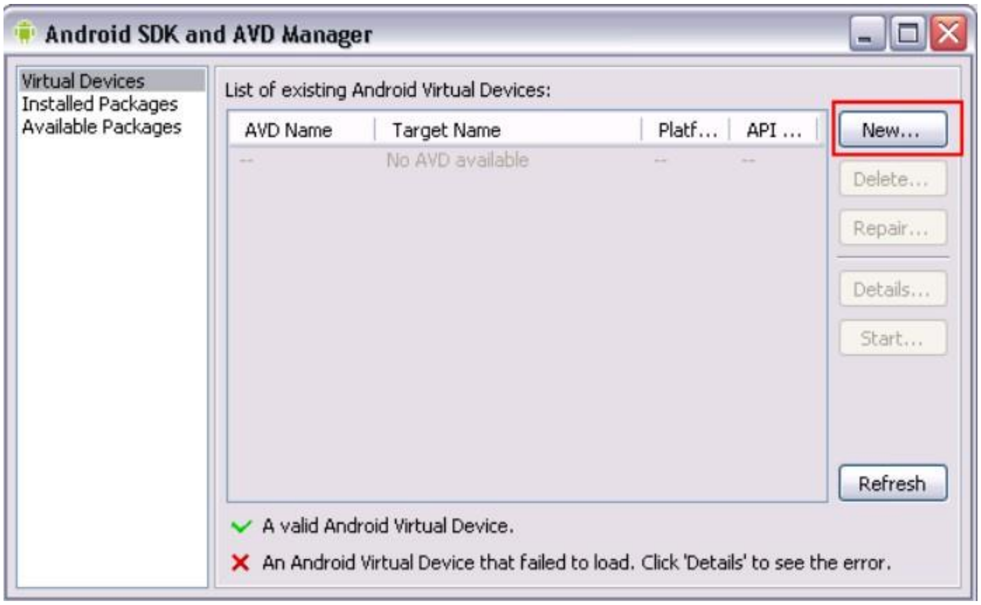
Haris Peco maintains plugins with provides access to the Android Source code. Use the Eclipse update manager to install two of his plugins.

Create an Android Emulator Device

The Android tools include an emulator. This emulator behaves like a real Android device in most cases and allow you to test your application without having a real device. You can emulate one or several devices with different configurations. Each configuration is defined via an "Android Virtual Device" (AVD).

To define an AVD press the device manager button, press "New" and maintain the following.


Plug-in Development - de.vogella.
File Edit Run Source Refactor Navigat

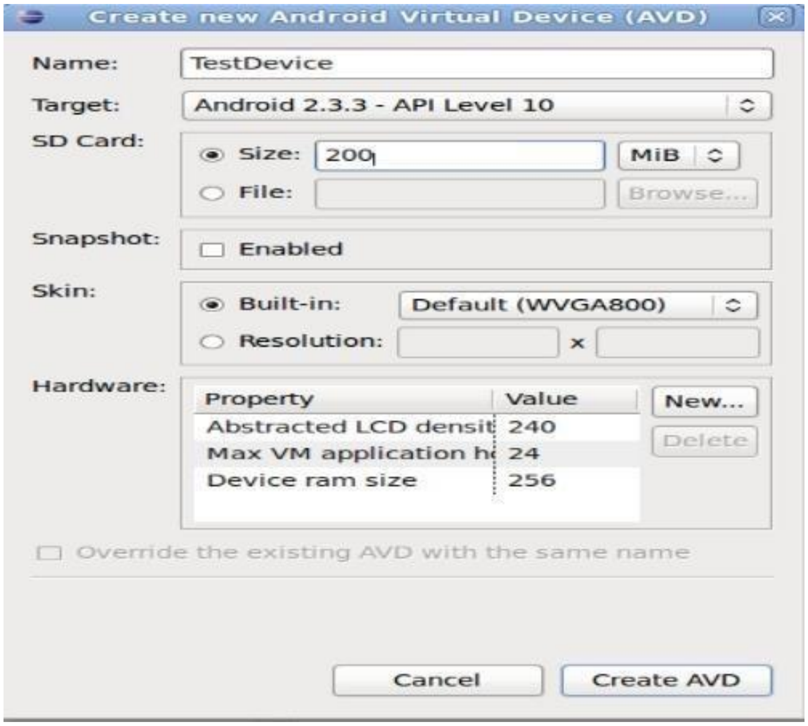


List of existing Android Virtual Devices:

AVD Name	Target Name	Platf...	API ...
--	No AVD available	--	--

✓ A valid Android Virtual Device.

✗ An Android Virtual Device that failed to load. Click 'Details' to see the error.



Name: TestDevice

Target: Android 2.3.3 - API Level 10

SD Card: ● Size: 200 MiB ○ File:

Snapshot: □ Enabled

Skin: ● Built-in: Default (WVGA800) ○ Resolution: x

Hardware:

Property	Value
Abstracted LCD densit	240
Max VM application h	24
Device ram size	256

□ Override the existing AVD with the same name

Press "Create AVD". This will create the device and display it under the "Virtual devices". To test if your setup is correct, select your device and press "Start".

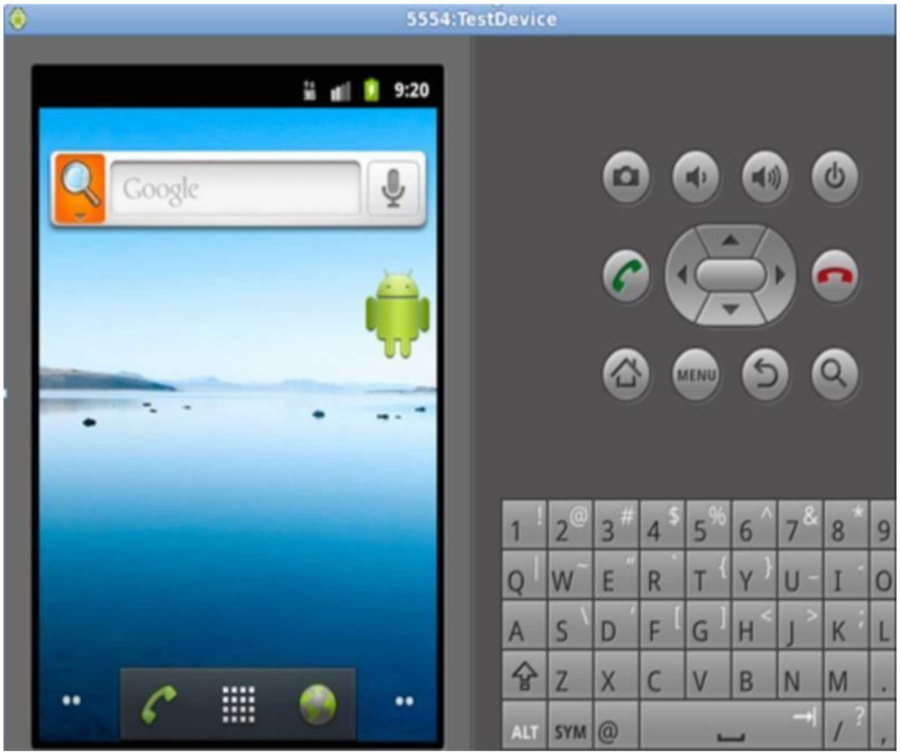


Fig. 4.3: AVD Device

4. Error handling

Things are not always working as they should be. Several users report that get the following errors:

Project ... is missing required source folder: 'gen'

The project could not be built until build path errors are resolved.

Unable to open class file R.java.

To solve this error, select from the menu Project -> Clean.

If you having problems with your own code, you can use the Log Cat viewer as described in Log Cat Viewer.

CHAPTER 5

SYSTEM ANALYSIS AND DESIGN

Systems development is a systematic process which includes phases such as planning, analysis, design, deployment, and maintenance. System Analysis is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components. System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose. Analysis specifies what the system should do.

System Design is a process of planning a new business system or replacing an existing system by defining its components or modules to satisfy the specific requirements. Before planning, you need to understand the old system thoroughly and determine how computers can best be used in order to operate efficiently. System Design focuses on how to accomplish the objective of the system.

5.1 UML DIAGRAMS:

UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: A Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modelling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems.

The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modelling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modelling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

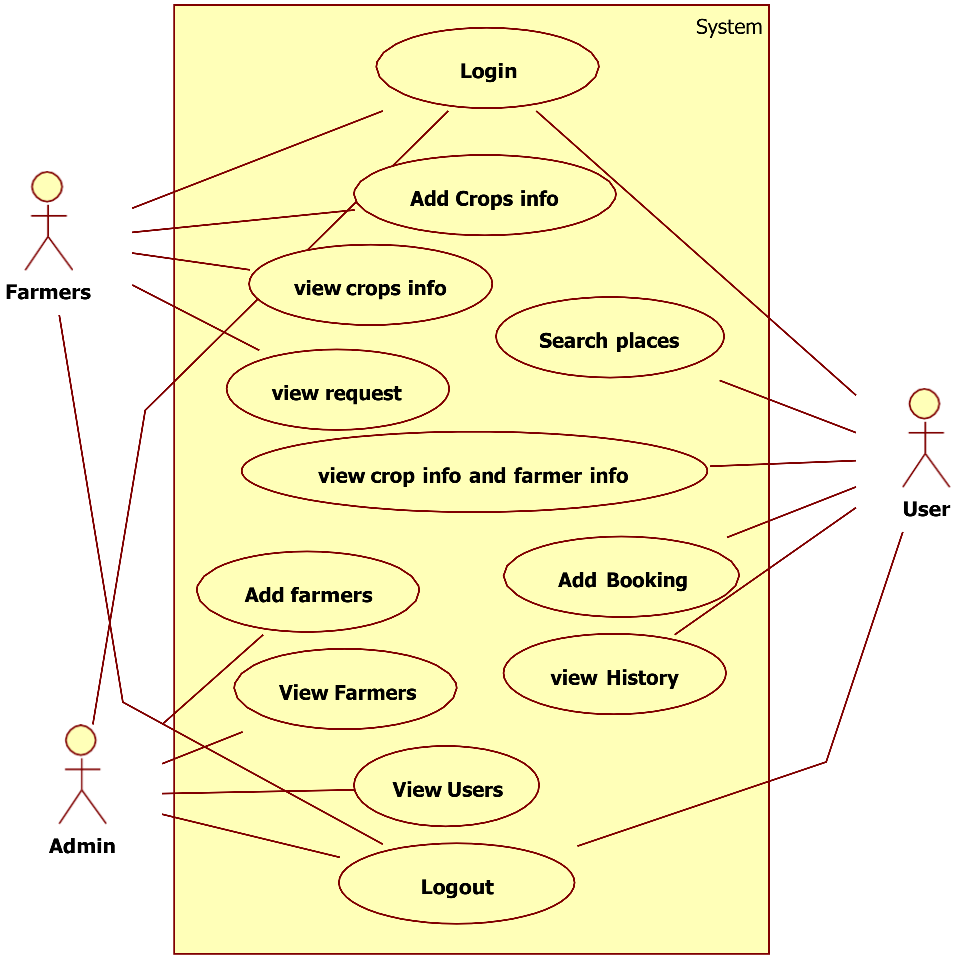


Fig. 5.1: Use case Diagram

CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

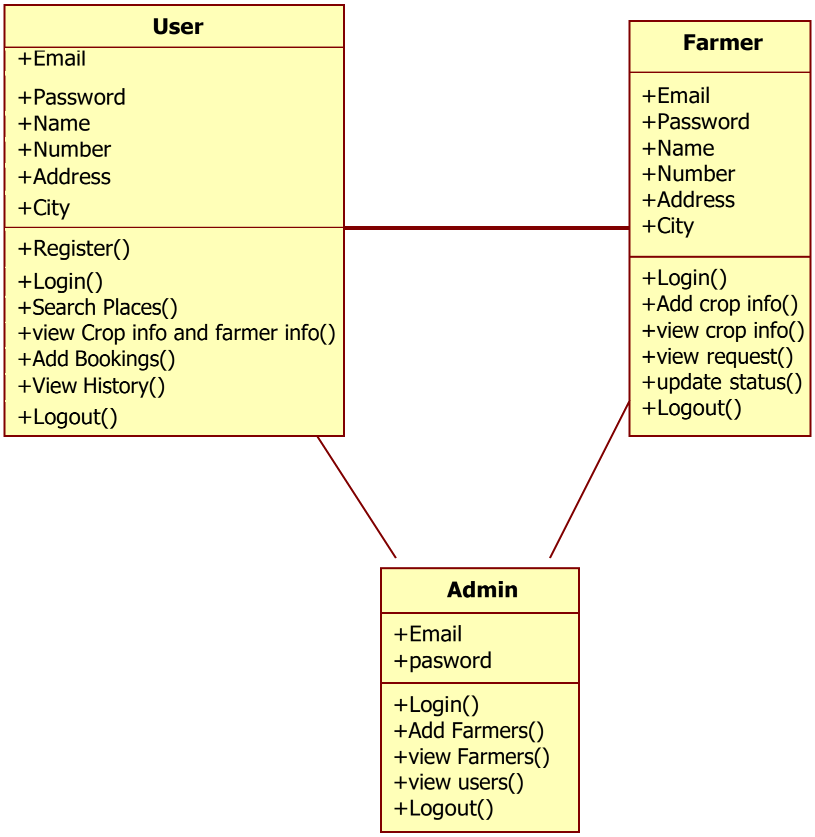


Fig. 5.2: Class Diagram

SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

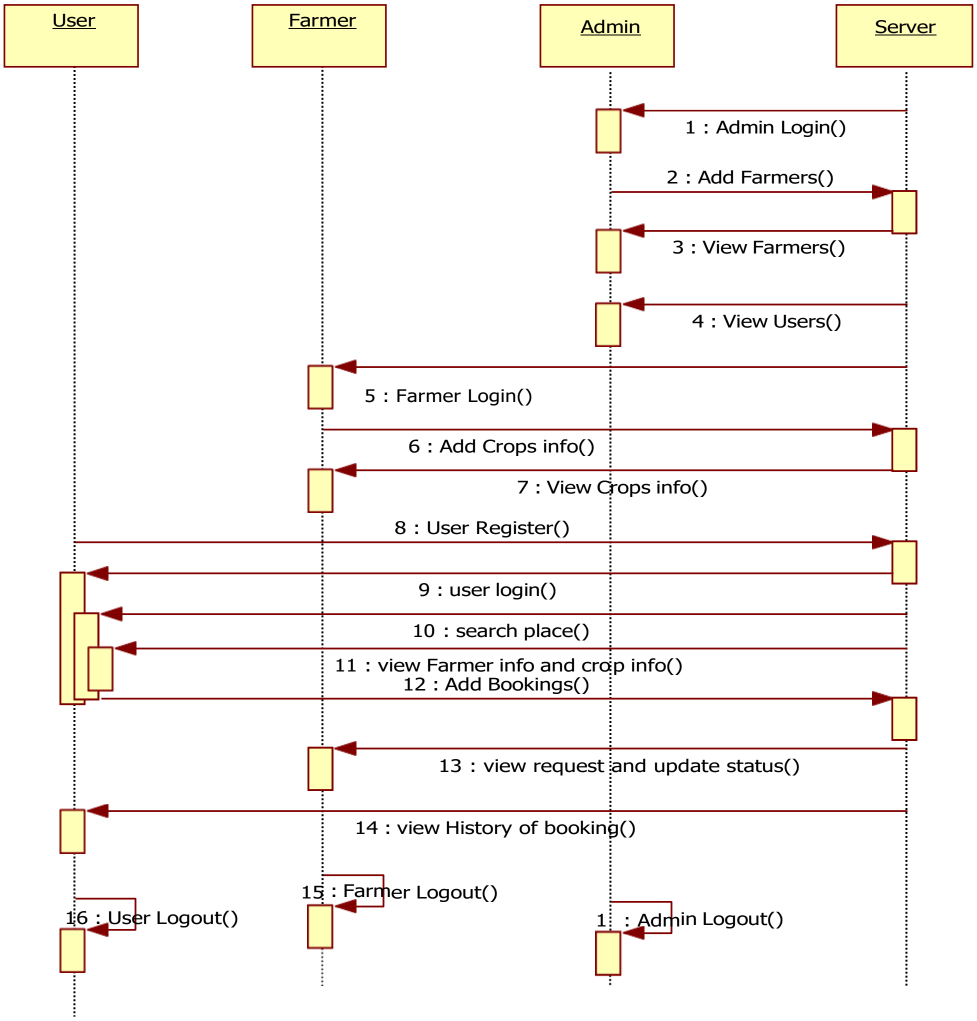


Fig. 5.3**:** Sequence Diagram

COLLABORATION DIAGRAM:

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization where as the collaboration diagram shows the object organization.

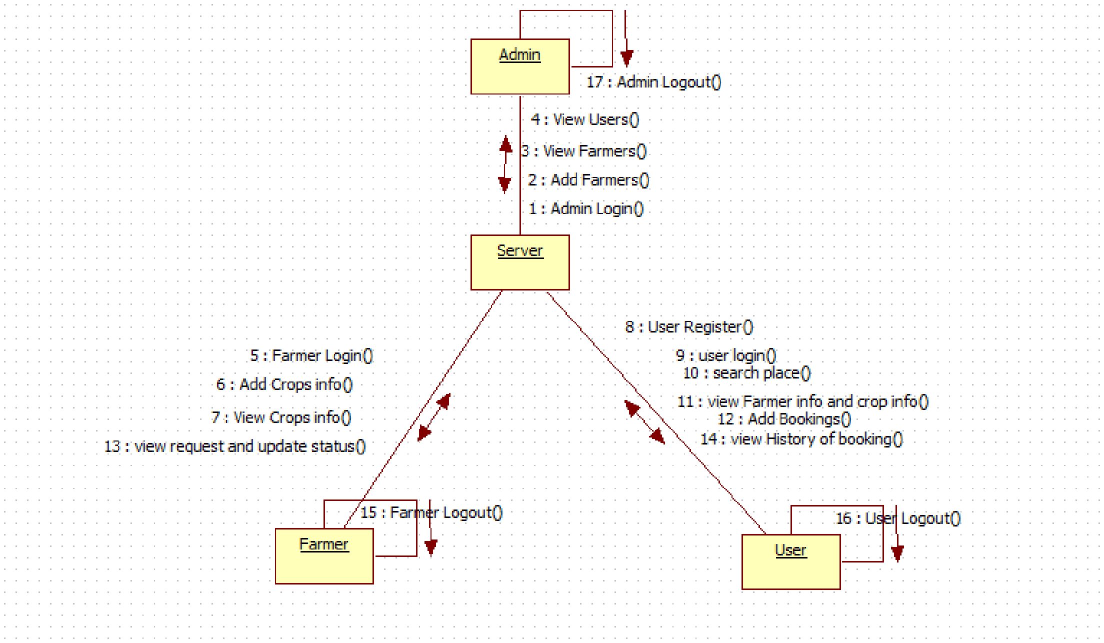


Fig. 5.4: Collaboration Diagram

ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

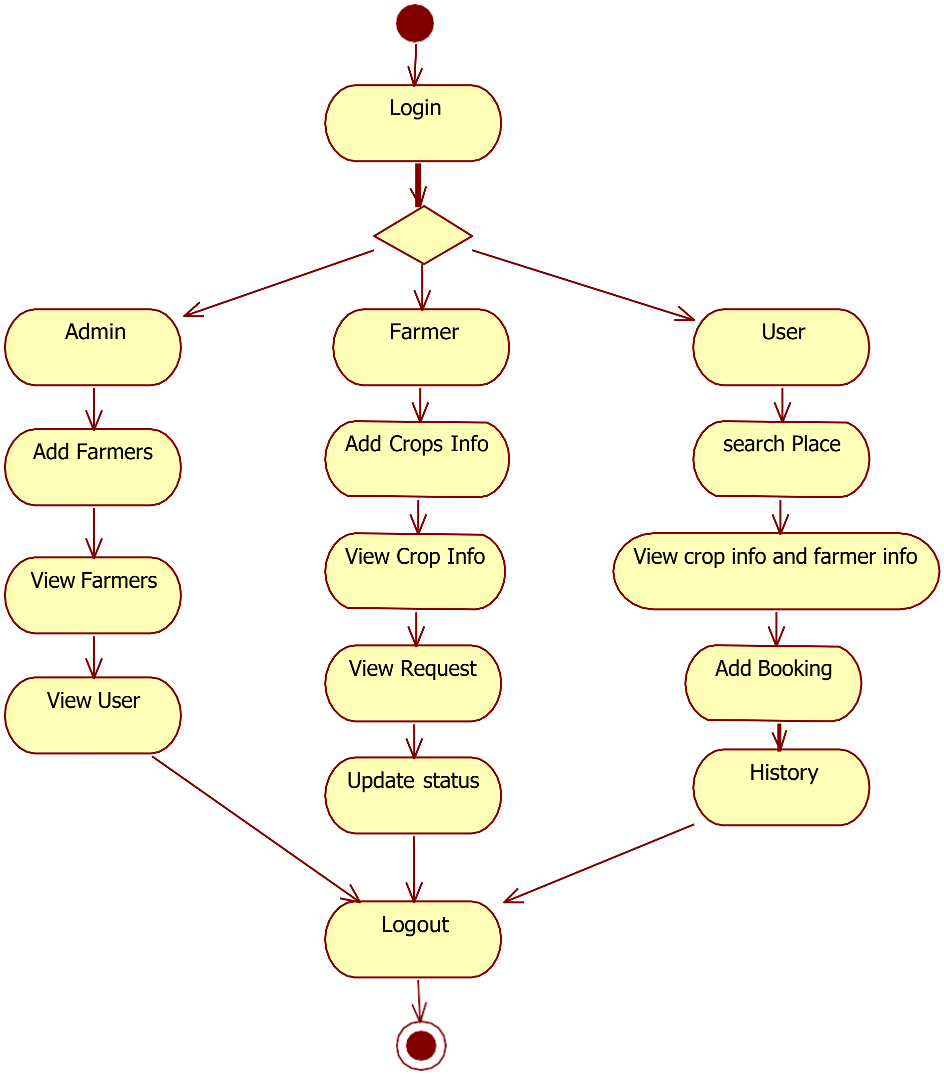


Fig. 5.5: Activity Diagram

Component Diagram:

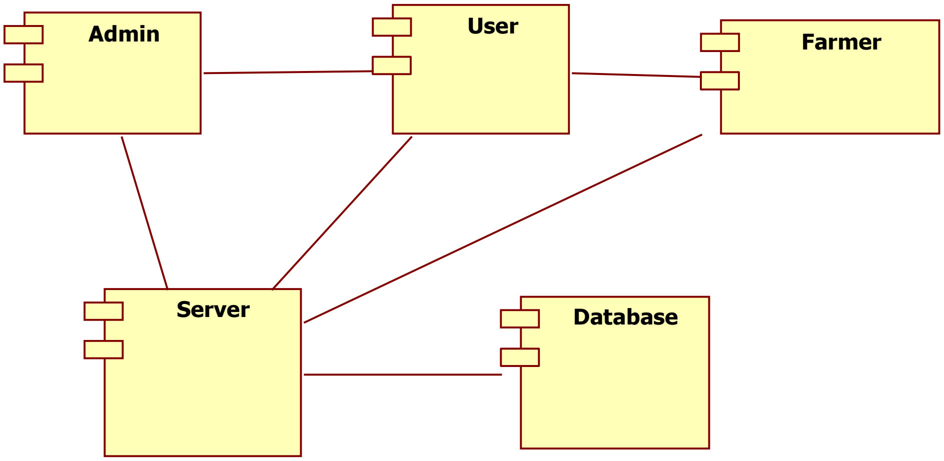


Fig. 5.6: Component Diagram

Deployment Diagram:

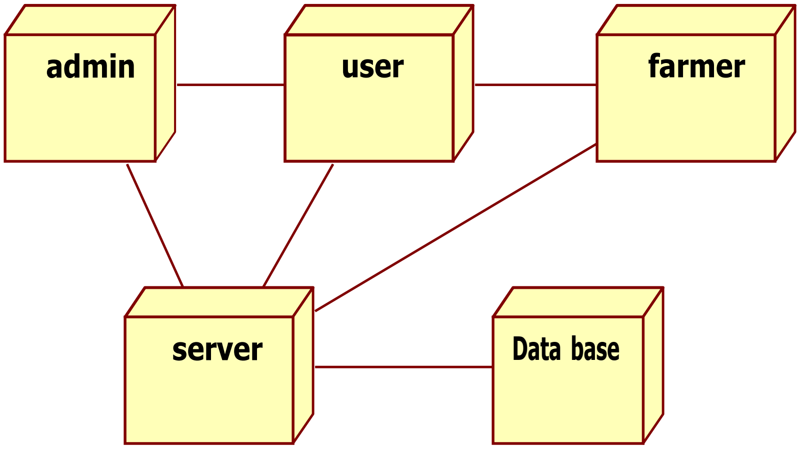


Fig. 5.7**:** Deployment Diagram

ER Diagram:

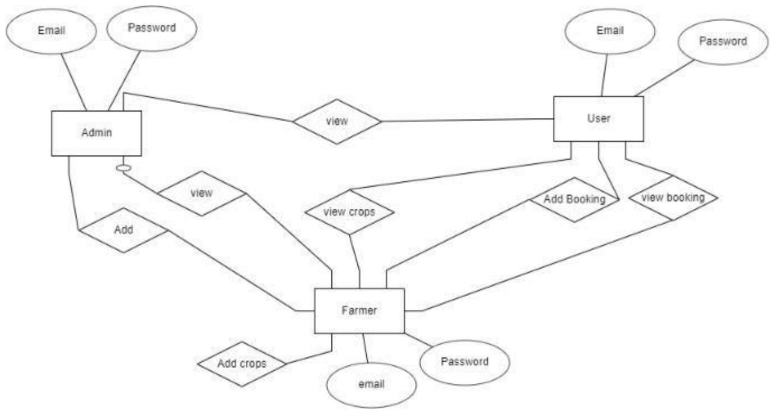


Fig. 5.8: ER Diagram

DFD Diagram:

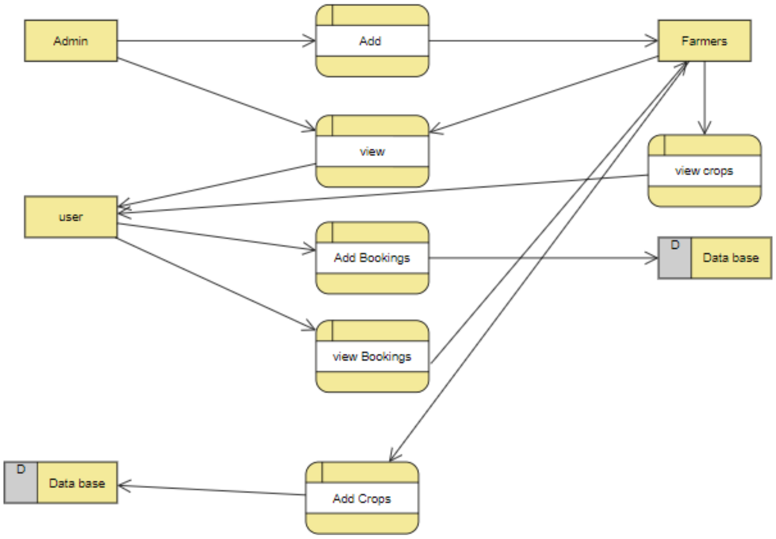


Fig. 5.9: DFD Diagram

CHAPTER 6

IMPLEMENTATION

6.1 Module And Functionalities

Admin: He will login with his email and password. He can add the farmers and view farmers.

Farmer: He will login with email and password. He will add the crop information and view the tourism people. He can view history of the crop information also.

User: He will register by providing some details after that the user can login with his email and password. The user view crop information and farmer details then he can book a slot with farmers and view the status and location of that farmers.

6.2 Architecture:

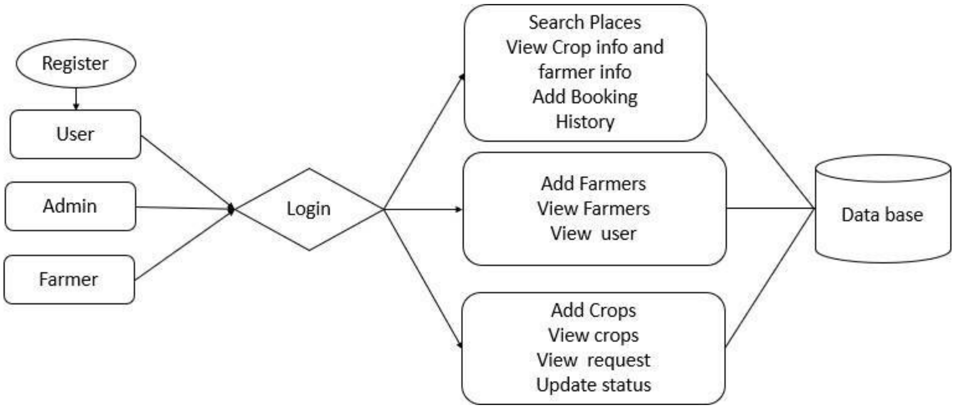


Fig. 6.1: Architecture Diagram

6.3 Data Dictionary:

A data dictionary contains metadata i.e., data about the database. The data dictionary is very important as it contains information such as what is in the database, who is allowed to access it, where is the database physically stored etc. The users of the database normally don't interact with the data dictionary, it is only handled by the database administrators.

The data dictionary in general contains information about the following

- Names of all the database tables and their schemas.
- Details about all the tables in the database, such as their owners, their security constraints, when they were created etc.
- Physical information about the tables such as where they are stored and how.
- Table constraints such as primary key attributes, foreign key information etc.
- Information about the database views that are visible.

CHAPTER 7

TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

TYPES OF TESTS

7.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

7.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

7.3 FUNCTIONAL TEST

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centred on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

7.4 SYSTEM TESTING

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

7.5 WHITE BOX TESTING

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

7.6 BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

7.7 ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No encountered.

CHAPTER 8

RESULTS

In the final implementation of the application the first screen the user can view is the logo of android application.

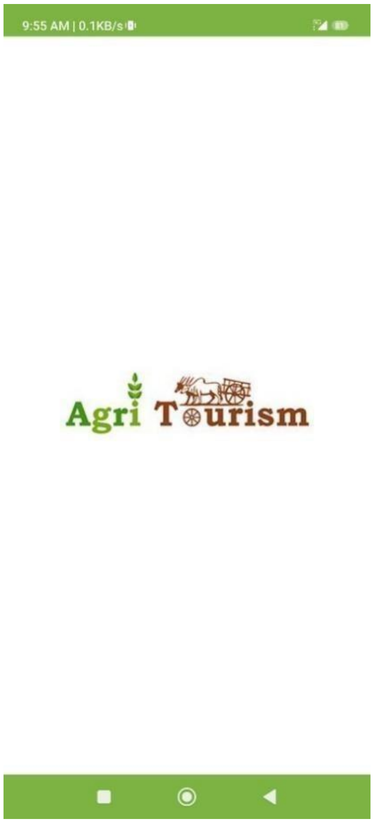


Fig. 8.1: Home Page

The above figure represents the initial screen of application.

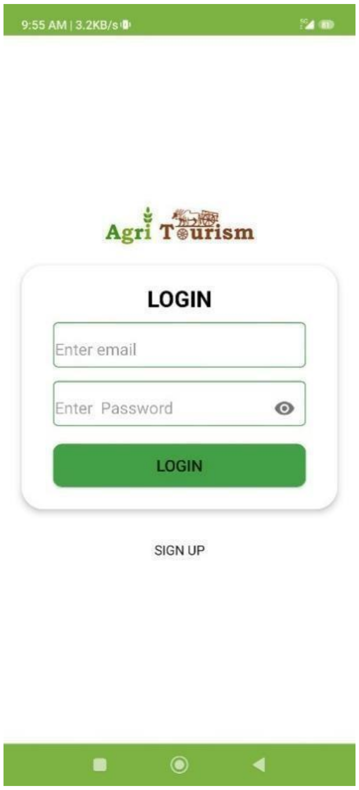


Fig. 8.2: Login Page

The above figure represents the Login Page of Admin. In this page, Admin will login with Email and Password.

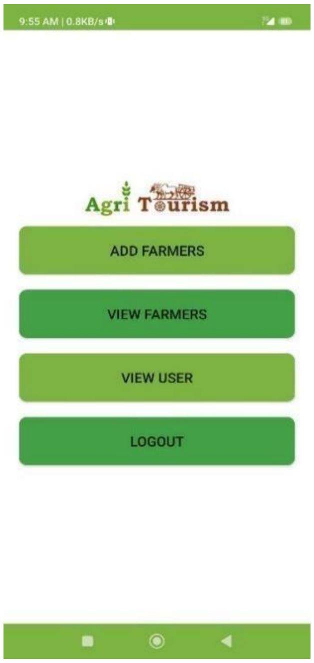


Fig. 8.3 : Admin Dashboard

The above page represents the admin dash board. In this page, admin can add farmers, view farmers, view users, and he can logout also.

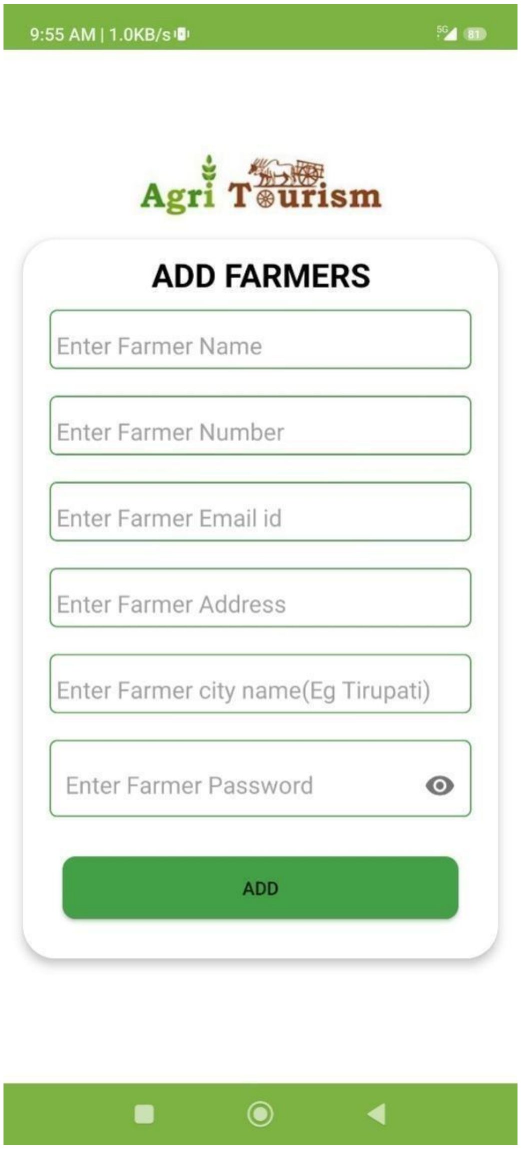
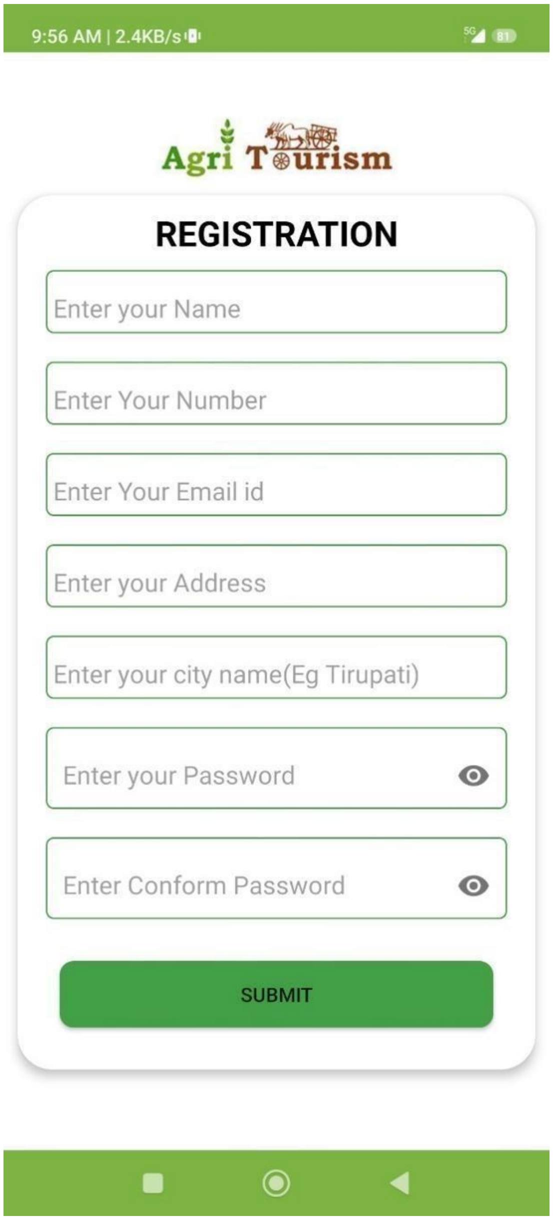


Fig. 8.4: Add Farmers Page

The above page is the Add Farmers page of application. Here, admin will add the farmers through their details and he will provide the login credentials for farmers.



9:56 AM | 2.4KB/s

Agri Tourism

REGISTRATION

Enter your Name

Enter Your Number

Enter Your Email id

Enter your Address

Enter your city name(Eg Tirupati)

Enter your Password

Enter Conform Password

Fig. 8.5: User Registration Page

The above page is the User Registration page. In this, user will get register by providing their details.

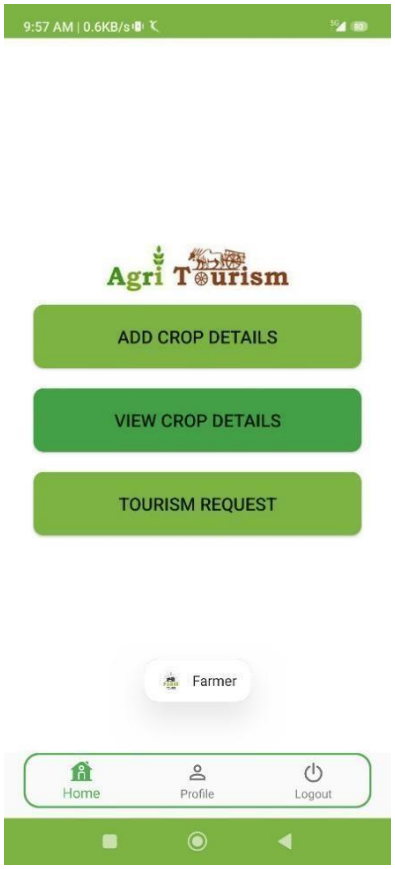


Fig. 8.6: Farmer Dashboard

The above page is the Farmer dash board page where farmer will add crop details, view crop details and he can view tourism request.

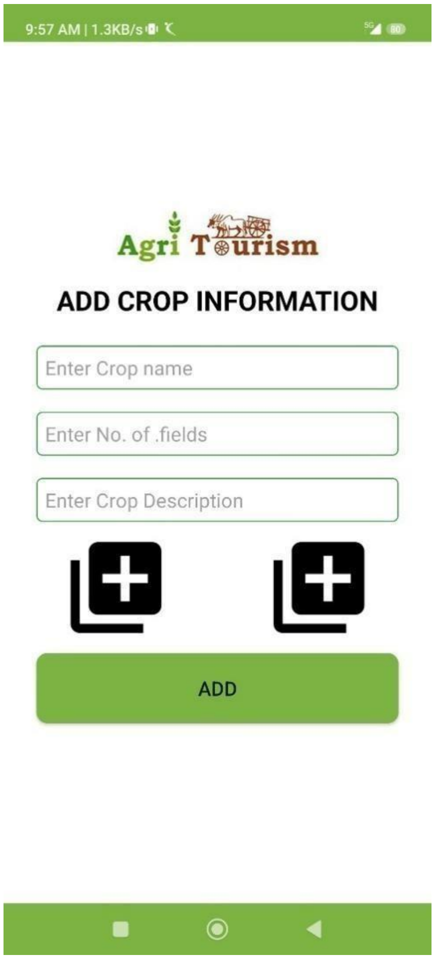
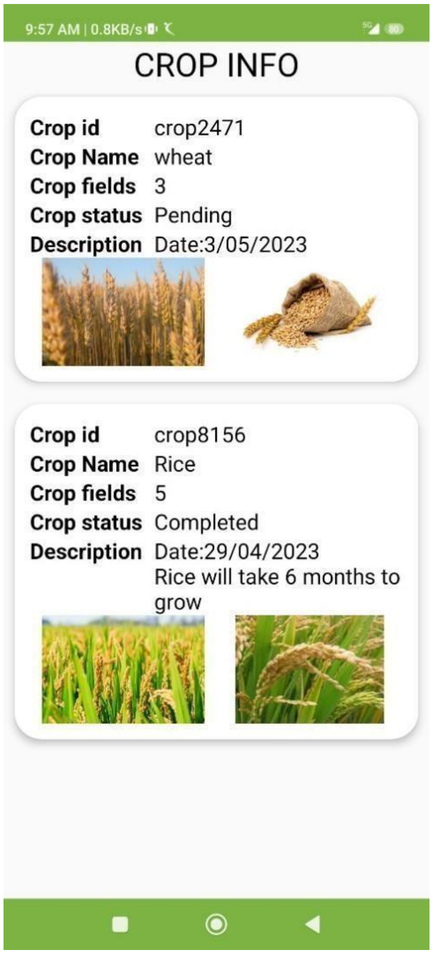


Fig. 8.7: Add crop information page

The above page is Add crop Information page where farmers will add the crop details such as crop name, number of fields that the crop has sown and crop description.



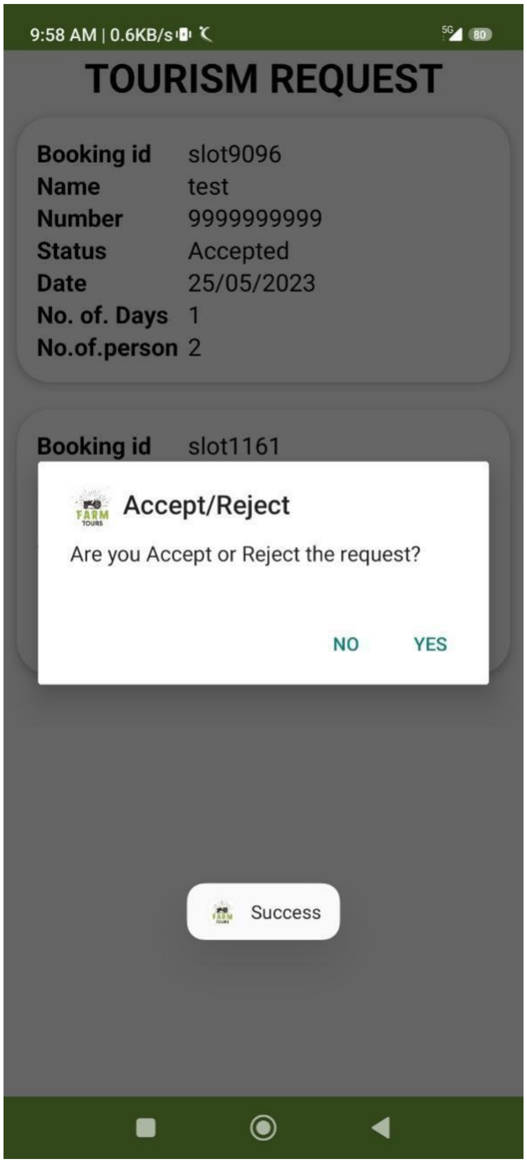
CROP INFO

Crop id crop2471
Crop Name wheat
Crop fields 3
Crop status Pending
Description Date:3/05/2023

Crop id crop8156
Crop Name Rice
Crop fields 5
Crop status Completed
Description Date:29/04/2023
Rice will take 6 months to grow

Fig. 8.8**:** Crop Info Page

The above page will result the crop Information which contain crop id, crop name, crop fields, crop status and description.



9:58 AM | 0.6KB/s

TOURISM REQUEST

Booking id	slot9096
Name	test
Number	9999999999
Status	Accepted
Date	25/05/2023
No. of. Days	1
No.of.person	2

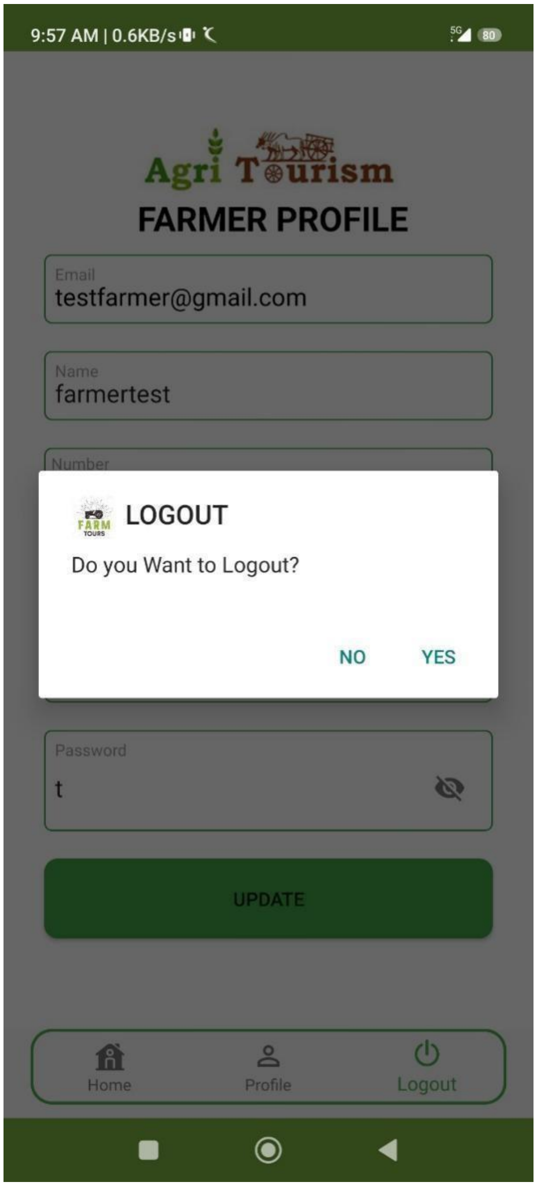
Booking id slot1161

Accept/Reject

Are you Accept or Reject the request?

Fig. 8.9: Tourism Request Page

The above page will get displayed for the farmers. So that farmer can view their request and may accept or reject the request according to their schedule.



FARMER PROFILE

Email
testfarmer@gmail.com

Name
farmertest

Number

LOGOUT

Do you Want to Logout?

Password
t

Fig. 8.10**:** Logout Page

The above page is logout page. Once the farmer is done with the application, he may logout from the app.

CONCLUSION AND FUTURE WORK

Agri-tourism is an emerging trend in the travel industry that offers visitors an opportunity to experience and learn about agriculture and farming practices. It is a unique way to connect with nature, learn about local cultures, and support the local economy. Overall, agri-tourism has numerous benefits for both the visitors and the local communities. Visitors get a chance to learn about agriculture, experience rural life, and enjoy fresh and organic food. At the same time, local communities benefit from the additional income generated from tourism, which can help sustain and promote their traditional farming practices. However, it is important to ensure that Agri-tourism is developed and managed in a sustainable and responsible way. This means preserving natural resources, protecting the environment, and respecting the cultural values of the local communities. It is also crucial to ensure that the farmers and the local communities are involved in the decision-making process and that the benefits of agri-tourism are shared fairly. Agri-tourism has great potential as a sustainable form of tourism that can provide a unique and authentic travel experience while supporting local communities and preserving the natural and cultural heritage of the destinations.

In the future, the agri-tourism project can focus on diversifying experiences by offering farm-to-table dining, cooking classes, and guided farm tours. Engaging local communities through training programs and showcasing their culture can enhance the authenticity of the experience. Targeted marketing strategies can attract a diverse range of visitors, while research can assess the project's impact and identify areas for improvement.

REFERENCES

[1] Smith, A., et al. (2021). "Mobile Technology in Agriculture: A Review of Its Impact on Farming Practices." Journal of Agricultural Informatics, 12(3), 45-60.

[2] Jones, S., & Williams, H. (2019). "Data-Driven Agriculture: Tools for Smallholder Efficiency." Global Food Security, 8(2), 34-42.

[3] Sunder Reddy, K. S. ., Lakshmi, P. R. ., Kumar, D. M. ., Naresh, P. ., Gholap, Y. N. ., & Gupta, K. G. . (2024). A Method for Unsupervised Ensemble Clustering to Examine Student Behavioral Patterns. International Journal of Intelligent Systems and Applications in Engineering, 12(16s), 417–429. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/4854.

[4] Farm-to-Table Initiative. (2018). "Bridging the Gap: Connecting Consumers and Farmers." Agriculture Today, 17(7), 19-25.

[5] Hussan, M.I. & Reddy, G. & Anitha, P. & Kanagaraj, A. & Pannangi, Naresh. (2023). DDoS attack detection in IoT environment using optimized Elman recurrent neural networks based on chaotic bacterial colony optimization. Cluster Computing. 1-22. 10.1007/s10586-023-04187-4.

[6] EcoAgriTech. (2023). "Agri-tourism in the Digital Age: Opportunities and Challenges." Journal of Sustainable Tourism, 11(6), 75-89.

[7] Wetland Preservation Society. (2024). "Digital Tools for Wetland Conservation: A Gap Analysis." Environmental Conservation Journal, 16(2), 112-127.

[8] Nagesh, C., Chaganti, K.Chaganti, S. , Khaleelullah, S., Naresh, P. and Hussan, M. 2023. Leveraging Machine Learning based Ensemble Time Series Prediction Model for Rainfall Using SVM, KNN and Advanced ARIMA+ E-GARCH. *International Journal on Recent and Innovation Trends in Computing and Communication*. 11, 7s (Jul. 2023), 353–358. DOI:https://doi.org/10.17762/ijritcc.v11i7s.7010.

An Android Application for Enhancing Agri-Tourism and Wetland Conservation through Farmer-Consumer Engagement

G. Nagaleela
Assistant Professor
Dept of CSE
Srinivasa Ramanujan Institute of Technology
Anantapur
Nagaleelag.cse@srit.ac.in

P. Gousiya
UG Scholar
Dept of CSE
Srinivasa Ramanujan Institute of Technology
Anantapur
gousiyaporalla@gmail.com

B. Anand
UG Scholar
Dept of CSE
Srinivasa Ramanujan Institute of Technology
Anantapur
204g1a0511@srit.ac.in

C. Manoj Reddy
UG Scholar
Dept of CSE
Srinivasa Ramanujan Institute of Technology
Anantapur
204g1a0554@srit.ac.in

K. Bramha Teja
UG Scholar
Dept of CSE
Srinivasa Ramanujan Institute of Technology
Anantapur
204g1a0524@srit.ac.in

ABSTRACT: In recent years, the agricultural sector has faced numerous challenges, including inadequate capacity to host agri-tourists, limited processing of viewable products, and a scarcity of knowledgeable practitioners. These barriers hinder the growth and development of agri-tourism, which is crucial for sustainable development and conservation efforts. Sustainable tourism has emerged as a significant force in conserving agricultural wetlands, bolstering local economies by creating jobs, and enhancing the income of those working in proximity to wetlands. In response to these challenges, this article introduces a novel initiative to develop an Android application aimed at bridging the communication gap between urban consumers and rural farmers. The app is designed to facilitate direct interaction, promote agricultural awareness, and enable knowledge exchange in crop science. Through this platform, users can gain insights into agricultural practices by virtually visiting farms, thus fostering community engagement and supporting agri-tourism. This system not only aims to improve the conservation of agricultural wetlands but also strives to elevate the socio- economic status of farmers by integrating them into the value chain of agri-tourism. The dual objectives of creating an efficient digital tool and promoting sustainable tourism through enhanced agri-tourism practices underscore the potential of technology in achieving economic sustainability and environmental conservation in rural landscapes.

Keywords: Agri-tourism, Wetlands Conservation, Sustainable Tourism, Agricultural Commodities, Android App, Knowledge Exchange, Community Engagement.

I. INTRODUCTION

In the evolving landscape of global agriculture, a significant chasm has developed between urban consumers and rural farmers. This disconnect is multifaceted, rooted in geographical distance, a lack of mutual understanding, and inadequate communication channels. Urban consumers often lack visibility into the origin of their food, the challenges of sustainable farming, and the conservation efforts required to maintain agricultural biodiversity, particularly in wetland areas. Conversely, rural farmers are frequently isolated from their potential marketplaces, lacking the tools to directly reach the consumers who are increasingly interested in the provenance and sustainability of their food.

The significance of bridging this gap extends beyond the simple transactional relationship of buyer and seller. It is a matter of empowering consumers to make informed choices that support sustainable farming practices and contribute to the conservation of critical ecosystems. For farmers, it represents the opportunity to gain fair prices for their commodities, share their knowledge and practices, and secure their livelihoods in a competitive global market. By connecting these two disparate groups, there is potential to foster a more sustainable, informed, and equitable food system [7].

The solution proposed is the development of an innovative Android application designed specifically to address this divide. The application aims to create a seamless and efficient platform that notifies urban consumers about local agricultural offerings, allows them to experience agri- tourism virtually, and promotes awareness about the importance of wetland conservation. This digital tool will serve as a conduit for knowledge exchange, bringing to light the intricacies of crop science and the day-to-day realities of rural farming [8]. Through this app, users can virtually visit farms, engage with farmers, and learn about the origins of their food, thereby fostering a sense of community and stewardship towards the environment. This initiative stands at the intersection of technology and agriculture, poised to redefine the landscape of agri-tourism and contribute to the larger goal of sustainable development [17-19]

1.1 **The Urban-Rural Divide:** The gap between urban consumers and rural producers is more than geographical, it is a divide characterized by a lack of communication and a dearth of understanding. City occupiers are often removed from the agrarian challenges and the intricate processes that bring food to their tables. Meanwhile, farmers, stewards of the land, find themselves marginalized in the narratives that shape consumer choices and policies. This disconnect hinders not only the economic prospects of farmers but also the broader societal move towards sustainable consumption.

1.2 **Impacts on Sustainability and Conservation:** This disconnect reverberates through the sustainability of farming practices and the conservation of natural resources, particularly wetlands, which are often integral to agricultural ecosystems [10]. The lack of consumer awareness about the importance of these ecosystems and the role of sustainable agriculture in their preservation can lead to undervaluation of conservation efforts. Furthermore, farmers without a platform to share their sustainable practices may succumb to economic pressures, adopting less eco-friendly methods that jeopardize these delicate habitats [11].

1.3 **The Role of Technology in Agri-tourism:** Technology harbors the potential to revolutionize agri-tourism by providing immersive, informative experiences that bridge the gap between urban and rural spheres. Digital platforms can transport the urban populace to the very heart of agricultural operations, fostering a transparent and educational view of farming life and its reliance on the conservation of natural landscapes, such as wetlands [12].

1.4 **Innovative Solutions for Community Engagement:** To foster this connection, the proposed Android application serves as an innovative medium for community engagement. It is poised to enable urban users to engage with the rural agricultural world, facilitating a virtual exchange that can translate into real-world impact. Through interactive features, the app aims to deliver educational content, real-time farm experiences, and direct communication channels between consumers and farmers, nurturing a community around shared interests in sustainability and conservation [13].

Objective and Scope of the Application: The objective of this application is twofold: to enhance the visibility of rural agriculture within urban communities and to contribute actively to the conservation of agricultural wetlands [14]. The scope of the app extends beyond mere transactional

interactions; it encompasses the delivery of educational content on crop science, the promotion of sustainable farming practices, and the support of eco-friendly agri-tourism.

Anticipated Outcomes: The implementation of this Android app is anticipated to yield significant outcomes. For users, it promises an enriched understanding of and direct engagement with the agricultural process. For farmers, it provides a platform to showcase their practices, expand their market reach, and potentially improve their economic standing. Environmentally, it is expected to contribute to the conservation conversation, advocating for the protection of agricultural wetlands[20][21] and sustainable farming practices. Collectively, these outcomes embody the larger vision of a connected, informed, and sustainable agricultural future.

II. LITERATURE SURVEY

2.1 Review of Agricultural Apps and Their Impact: The proliferation of mobile technology has given rise to a myriad of agricultural applications aimed at enhancing farm management, market access, and consumer education. A survey of the current literature reveals a diverse landscape of digital tools that assist in various aspects of farm operation, from precision agriculture to supply chain logistics (Smith et al., 2021) [1]. These applications have had a measurable impact on the efficiency and output of farms, enabling farmers to make data-driven decisions (Jones & Williams, 2019). Moreover, apps designed for consumer use have been shown to increase awareness of sustainable practices and local food systems (GreenTech, 2020). However, few have managed to provide a comprehensive platform that addresses the needs of both farmers and consumers within the context of agri-tourism and wetland conservation [2].

2.2 Discussion of Previous Efforts to Connect Consumers with Farmers: Efforts to connect consumers directly with farmers have been well-documented. Community-supported agriculture (CSA) and farmers' markets have been at the forefront of this movement, creating spaces for direct purchase and dialogue (Farm-to-Table Initiative, 2018). Digital marketplaces and farm-to-fork apps have also emerged, offering platforms for farmers to sell their produce directly to consumers (AgriMarketplace, 2022) [3]. While these initiatives have successfully shortened the food supply chain, they often lack components of education and interactive engagement that are critical in fostering a deeper understanding and appreciation for agricultural processes and ecosystems, particularly among urban dwellers [4].

2.3 Identification of Gaps in Current Solutions: Despite the progress made, there remains a gap in solutions that integrate the experiential aspect of agri-tourism[15] with the functional utility of agricultural apps. Existing literature points to a need for a more integrated approach that combines the transactional ease of marketplace apps with the educational and experiential richness of agri-tourism (EcoAgriTech, 2023) [5]. Additionally, there is a significant void in applications that focus on the conservation of wetlands through agricultural activities (Wetland Preservation Society, 2024). These gaps suggest an opportunity for an application that not only facilitates the direct sale of farm commodities but also immerses the user in the agricultural experience and educates them on the importance of sustaining wetland ecosystems [6].

III. METHODOLOGY

3.1 App Development Process: The methodology for developing the Android application was comprehensive, encompassing user research, iterative design, and rigorous testing to ensure functionality and ease of use. User research began with a series of focus groups and surveys targeting both potential users and farmers to ascertain their needs, preferences, and technological capabilities. This initial research informed the design phase, which utilized the Agile software development framework, allowing for flexible and adaptive construction of the application with regular feedback loops. Prototyping was an integral part of the design process, with wireframes and mock-ups created to visualize the app's features. Subsequent testing methodologies included a combination of alpha and beta testing with both target user groups to refine the user interface (UI) and user experience (UX) design. Functional testing, usability testing, and performance testing were conducted to ensure the app's reliability, efficiency, and scalability.

3.2 Strategies for User-Farmer Interaction: To facilitate seamless and efficient interaction between users and farmers, the app incorporates a real-time messaging system, allowing users to send queries and receive responses from farmers directly. A user-friendly interface prioritizes accessibility, ensuring that users of varying technological proficiency can navigate the app with ease. The app also includes a feature that allows users to schedule virtual farm visits, providing a live video feed from the farm, thus simulating a real agri-tourism experience. To further enhance this interaction, the app integrates a translation feature to overcome language barriers, thereby broadening the scope of user-farmer communication.

Methodology
App Development Process
Strategies for User-Farmer Interaction
Promotion of Agricultural Awareness
Community Engagement

Fig 1: Methodology

3.3 Promotion of Agricultural Awareness and Community Engagement: To promote agricultural awareness, the app features an educational portal with resources on crop science, sustainable farming practices, and the importance of wetland conservation. The content is curated to cater to a diverse audience, ranging from individuals with no prior agricultural knowledge to those seeking advanced understanding. Community engagement is stimulated through the app’s community forum, where users and farmers can discuss topics of mutual interest, share experiences, and organize community-driven conservation efforts. Gamification elements such as achievements and informational quests are incorporated to encourage users to engage with educational content and participate in community initiatives actively.

IV. DEVELOPMENT PROCESS

4.1 Technical Aspects of the App: The application was developed for the Android platform, chosen for its widespread use and open-source nature, providing flexibility and a broad user base. The core development utilized Kotlin, the preferred language for Android app development, known for its safety features and concise syntax. Android Studio served as the integrated development environment (IDE), providing a robust suite of tools for efficient coding, debugging, and testing. For the backend, Firebase was implemented due to its scalability and suite of tools that facilitate real-time database management, user authentication, and hosting. Additionally, the app integrates Google Maps API for geolocation services, allowing users to locate farms and understand the geographical context of their food sources.

4.2 Design Principles for User-Friendly Interface: The design of the app's interface adhered to Material Design guidelines, ensuring consistency, intuitiveness, and aesthetic appeal across the application. A user-centric approach was paramount, which involved iterative user testing to refine the UI/UX. Emphasis was placed on minimizing user input and ensuring that essential functions were accessible within a few taps. Large, legible typography and color contrasts were utilized to enhance readability, while interactive elements such as buttons and links were designed to be easily navigable, even on smaller devices. The design also accommodated accessibility features, like screen readers and voice commands, to cater to a diverse user demographic with varying levels of tech proficiency.

4.3 Features Bridging the Urban-Rural Divide: To address the urban-rural divide, the app introduced several key features. A 'Farm Discovery' feature allows users to explore farms through a virtual tour, offering 360-degree views of farms and detailed information about farming practices and crop types. A 'Direct Farmer Chat' function was incorporated to enable real- time conversations between consumers and farmers, facilitating a transparent dialogue and exchange of knowledge. To further educational goals, an 'Agricultural Learning Hub' was integrated, providing articles, videos, and interactive content related to sustainable farming, wetland conservation, and agri-tourism opportunities. Additionally, the app included a 'Community Events' feature, which announces local agri- tourism events, workshops, and conservation activities, encouraging user participation and fostering a sense of community around agricultural stewardship.

V. EVALUATION

5.1 Measuring the App's Impact on Users and Farmers: The evaluation of the app's impact involves a multi- dimensional approach, focusing on both qualitative and quantitative metrics. User engagement will be measured through data analytics, tracking metrics such as the number of app downloads, active user rates, frequency of use, and interaction rates with various features (e.g., 'Farm Discovery', 'Direct Farmer Chat'). Feedback surveys and user reviews will be utilized to gather qualitative insights into user satisfaction, usability, and areas for improvement. For farmers, the impact will be assessed by measuring changes in direct consumer engagement, increased inquiries and visits, and any reported changes in sales or revenue streams. Additionally, interviews and surveys with participating farmers will provide qualitative data on their experiences and perceived benefits from app.

5.2 Preliminary Feedback and Testing Results: Initial testing results indicate a positive reception from both users and farmers. Beta testing, conducted with a small group of users and farmers, revealed that users appreciated the ease of access to agricultural information and the opportunity to directly interact with farmers. Farmers reported an increase in inquiries and interest in their products and practices. Usability testing highlighted the app's intuitive design, but also pointed to the need for enhanced interactive elements within the 'Agricultural Learning Hub'. These findings have been instrumental in refining the app for its public release.

5.3 Promotion of Knowledge Exchange and Community Engagement: The app is designed to foster a vibrant community around agriculture and sustainability. The 'Direct Farmer Chat' feature has been a vital tool for knowledge exchange, allowing users to learn directly from farmers about sustainable practices, crop cycles, and the importance of wetland conservation. The educational content within the 'Agricultural Learning Hub' has been crafted to engage users of all knowledge levels, providing a stepping stone for deeper understanding and involvement in sustainable agriculture and environmental stewardship. This multifaceted approach has not only informed users but also created a platform for shared learning and community-driven initiatives in sustainable farming and conservation.

VI. SAMPLE SCREENS

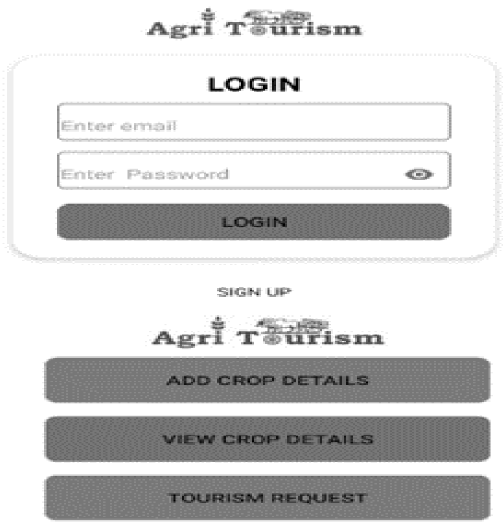


Fig 2:Home Screen

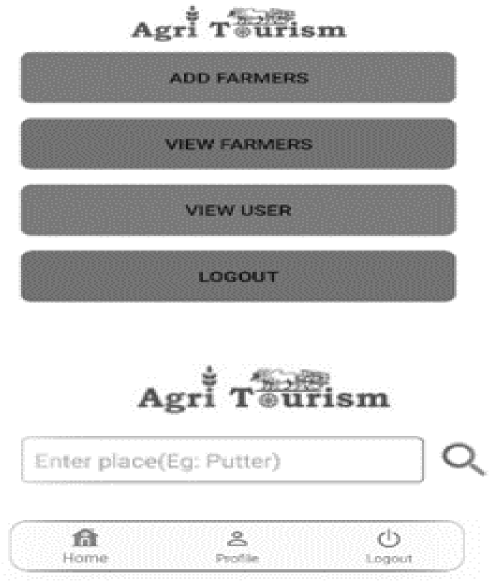


Fig 3: Search Screen

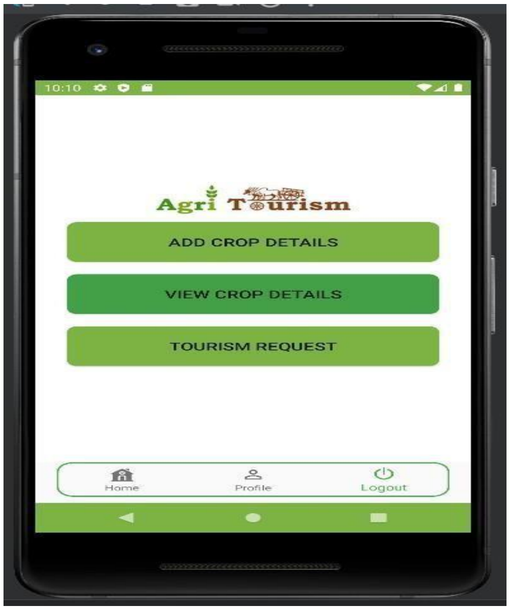


Fig 4:add/view and request page

Table 1: Results of existing and proposed

	Serving Rate (%)	**Accuracy (%)**
existing	86.24	92.36
proposed	88.91	93.69

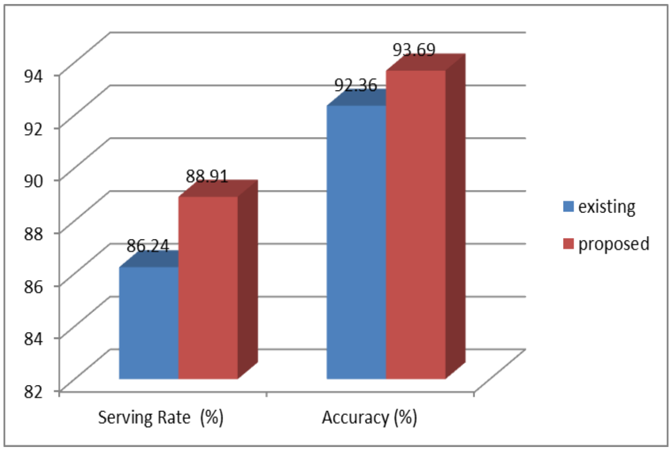


Fig 4: accuracy and serving rate of application

VII. CONCLUSION

In conclusion, this article has outlined the development and impact of an Android app aimed at bridging the urban-rural divide, highlighting its role in enhancing agri-tourism, promoting sustainable farming, and wetland conservation. Achieving its research objectives, the app has successfully facilitated meaningful interactions between consumers and farmers, fostering education and community engagement. Future development avenues include incorporating advanced technologies like AI and AR for enriched user experiences and expanding educational content. The app’s continuous evolution, informed by user feedback and technological advancements, epitomizes the potential of digital solutions in connecting communities, advancing sustainable practices, and nurturing a deeper appreciation for agricultural ecosystems.

REFERENCES

[1] Smith, A., et al. (2021). "Mobile Technology in Agriculture: A Review of Its Impact on Farming Practices." Journal of Agricultural Informatics, 12(3), 45-60.

[2] Jones, S., & Williams, H. (2019). "Data-Driven Agriculture: Tools for Smallholder Efficiency." Global Food Security, 8(2), 34-42.

[3] Sunder Reddy, K. S. ., Lakshmi, P. R. ., Kumar, D. M. ., Naresh, P. ., Gholap, Y. N. ., & Gupta, K. G. . (2024). A Method for Unsupervised Ensemble Clustering to Examine Student Behavioral Patterns. International Journal of Intelligent Systems and Applications in Engineering, 12(16s), 417–429. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/4854.

[4] Farm-to-Table Initiative. (2018). "Bridging the Gap: Connecting Consumers and Farmers." Agriculture Today, 17(7), 19-25.

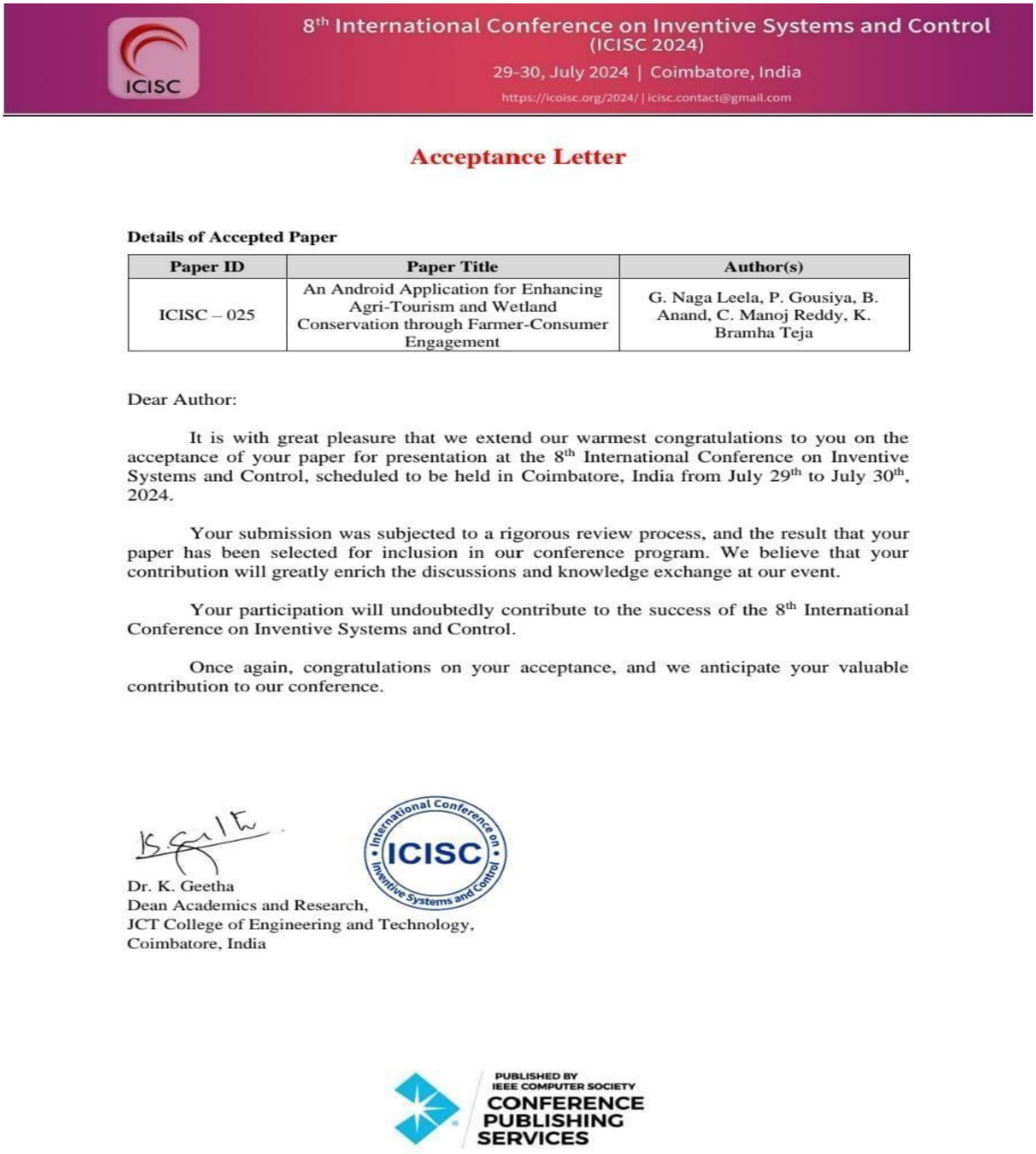
[5] Hussan, M.I. & Reddy, G. & Anitha, P. & Kanagaraj, A. & Pannangi, Naresh. (2023). DDoS attack detection in IoT environment using optimized Elman recurrent neural networks based on chaotic bacterial colony optimization. Cluster Computing. 1-22. 10.1007/s10586-023- 04187-4.

[6] EcoAgriTech. (2023). "Agri-tourism in the Digital Age: Opportunities and Challenges." Journal of Sustainable Tourism, 11(6), 75-89.

[7] Wetland Preservation Society. (2024). "Digital Tools for Wetland Conservation: A Gap Analysis." Environmental Conservation Journal, 16(2), 112-127. Nagesh, C., Chaganti, K.R. , Chaganti, S. , Khaleelullah, S., Naresh, P. and Hussan, M. 2023. Leveraging Machine Learning based Ensemble Time Series Prediction Model for Rainfall Using SVM, KNN and Advanced ARIMA+ E-GARCH. International Journal on Recent and Innovation Trends in Computing and Communication. 11, 7s (Jul. 2023), 353–358. DOI:https://doi.org/10.17762/ijritcc.v11i7s.7010.

[8] Naresh, P., & Suguna, R. (2021). Implementation of dynamic and fast mining algorithms on incremental datasets to discover qualitative rules. Applied Computer Science, 17(3), 82-91. https://doi.org/10.23743/acs-2021-23.

PAPER ACCEPTANCE LETTER



ICISC

8th International Conference on Inventive Systems and Control
(ICISC 2024)

29-30, July 2024 | Coimbatore, India

https://icoisc.org/2024/ | icisc.contact@gmail.com

Acceptance Letter

Details of Accepted Paper

Paper ID	Paper Title	Author(s)
ICISC – 025	An Android Application for Enhancing Agri-Tourism and Wetland Conservation through Farmer-Consumer Engagement	G. Naga Leela, P. Gousiya, B. Anand, C. Manoj Reddy, K. Bramha Teja

Dear Author:

It is with great pleasure that we extend our warmest congratulations to you on the acceptance of your paper for presentation at the 8th International Conference on Inventive Systems and Control, scheduled to be held in Coimbatore, India from July 29th to July 30th, 2024.

Your submission was subjected to a rigorous review process, and the result that your paper has been selected for inclusion in our conference program. We believe that your contribution will greatly enrich the discussions and knowledge exchange at our event.

Your participation will undoubtedly contribute to the success of the 8th International Conference on Inventive Systems and Control.

Once again, congratulations on your acceptance, and we anticipate your valuable contribution to our conference.

International Conference on Inventive Systems and Control
ICISC

Dr. K. Geetha
Dean Academics and Research,
JCT College of Engineering and Technology,
Coimbatore, India

PUBLISHED BY
IEEE COMPUTER SOCIETY
CONFERENCE
PUBLISHING
SERVICES