

UNDERGRADUATE PROJECT PROGESS REPORT

|  |  |
| --- | --- |
| **Project Title:** | **App-Recipe Management and Meal Planning System** |
| **Surname:** | **Lv** |
| **First Name:** | **Tao** |
| **Student Number:** | **202018010124** |
| **Supervisor Name:** | **Maged Refat** |
| **Module Code:** | **CHC 6096** |
| **Module Name:** | **Project** |
| **Date Submitted:** | **27/12/2023** |

**Table of Contents**

[1 Introduction 3](#_Toc154584160)

[1.1 Background 3](#_Toc154584161)

[1.2 Aim 3](#_Toc154584162)

[1.3 Objectives 4](#_Toc154584163)

[1.4 Project Overview 4](#_Toc154584164)

[1.4.1 Scope 4](#_Toc154584165)

[1.4.2 Audience 4](#_Toc154584166)

[2 Background Review 4](#_Toc154584167)

[3 Technical Progress 5](#_Toc154584168)

[3.1 Approach 5](#_Toc154584169)

[3.2 Technology 7](#_Toc154584170)

[3.3 Testing and Evaluation Plan 7](#_Toc154584171)

[3.4 Design and Implementation 9](#_Toc154584172)

[4 Project Management 10](#_Toc154584173)

[4.1 Activities 10](#_Toc154584174)

[4.2 Schedule 10](#_Toc154584175)

[4.3 Project Version Management 11](#_Toc154584176)

[4.4 Project Data Management 11](#_Toc154584177)

[4.5 Project Deliverables 12](#_Toc154584178)

[5 Professional Issues and Risk: 12](#_Toc154584179)

[5.1 Risk Analysis 12](#_Toc154584180)

[5.2 Professional Issues 13](#_Toc154584181)

[6 References 13](#_Toc154584182)

# Introduction

## Background

Eating a healthy diet can play a crucial role in preventing, delaying, or alleviating symptoms and complications associated with chronic illnesses, making it a vital aspect of managing these conditions [1]. Amiri and Li [2] have proposed that promoting a diet rich in fruits, vegetables, and whole grains while minimizing saturated fat intake can help individuals with heart disease reduce their risk of high blood pressure through meal planning. This underscores the significance of meal planning, particularly for those affected by such conditions. In other hand, meal planning can be challenging for individuals seeking to explore new cuisines or those with limited time for cooking, who are in need of more recipes tailored to their specific requirements. Punamiya [3] has identified a solution in the form of a recipe application that suggests recipes based on available ingredients, thereby reducing the time and cost associated with grocery shopping. The application contains a comprehensive database of recipes that users can browse, with the added flexibility of filtering recipes based on various criteria.

Managing meal planning and recipes involves handling substantial amounts of data. This poses a challenge to the security of data and the ability to process it. But now technology has made this challenge easier. As highlighted in Sharma's article [4], database management systems (DBMS) offer superior security and support concurrent access by multiple users, making them more advantageous than traditional file systems. Furthermore, mobile applications are convenient for users, requiring just a few taps on their smartphones, in contrast to web-based platforms.

Combining these insights, this project aims to develop a mobile-based DBMS that serves as a repository for storing recipes and facilitating meal planning. This system will benefit a wide range of users. Those interested in exploring new culinary experiences can easily search for diverse recipes, while individuals with time constraints can find recipes tailored to their specific time limitations. Additionally, users will have the ability to create meal plans for specific durations, with the system generating a corresponding shopping list based on the specified plan.

## Aim

Develop a database management system for storing recipes and planning meals on the mobile platform.

## Objectives

1. Complete background check of existing DBMS.
2. Complete detailed planning and design of various modules.
3. Implement the functions of each module.
4. Test each module using testing software.
5. Show works for audiences.

## Project Overview

### Scope

The software offers a platform to manage a wide array of recipes, allowing users to input, modify, and categorize them according to different factors like cuisine, dietary needs, ingredients, and preparation time. Additionally, users can devise meal schedules for specific periods, such as a week or a month, and generate a comprehensive shopping list based on their chosen recipes.

Once users install mobile software, they can open the software and use its features by clicking on the application icon. Mobile software can connect to the internet to obtain data or interact with other users, while also accessing device hardware (such as cameras, sensors, etc.) to provide richer functions and experiences.

### Audience

This app is perfect for individuals looking to take control of their diet. Whether you have a chronic condition or are simply focused on maintaining a healthy lifestyle, this app can help you manage your meals. Additionally, for those who enjoy trying new recipes, this app offers a wide variety of options to explore.

# Background Review

Professional companies have contributed existing technologies and resources to developers, such as the "Yon-Builder" [5] mobile development platform, which offers a powerful app engine to simplify app development. Additionally, "Developers" [6] provides training courses for Android applications, utilizing Jetpack package, Compose package, and Kotlin language. These resources have facilitated the successful creation of Android applications for many individuals. Leveraging these existing technologies, the development of this project will be streamlined and flexible, with strong scalability across various frameworks.

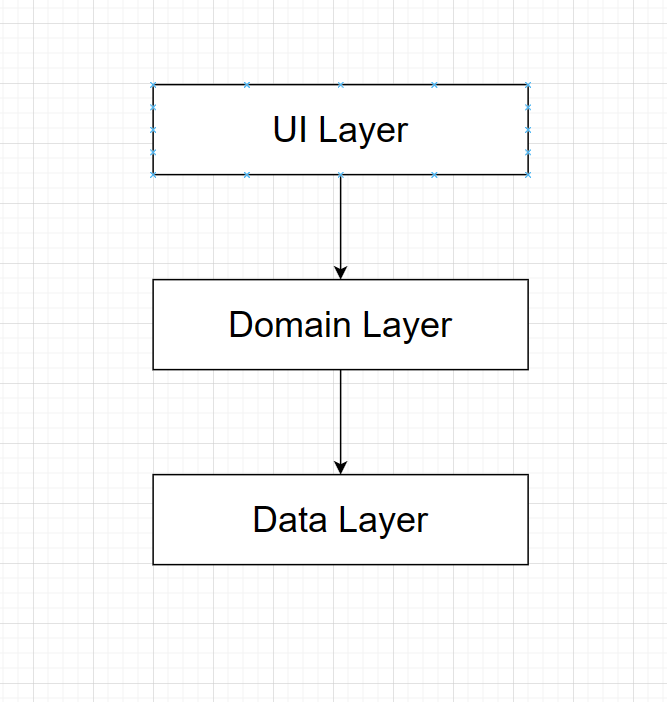
In today's market, there is a wide array of mobile applications available for Recipe Management and Meal Planning Systems. One such app, Paprika Recipe Manager [7], offers users access to a diverse range of delectable recipes. Furthermore, it allows for seamless synchronization of saved recipes across various devices, including computers, smartphones, and tablets. Additionally, the app features meal planning capabilities, the creation of well-organized shopping lists, and includes a timer function.

# Technical Progress

## Approach

Software Development Model:

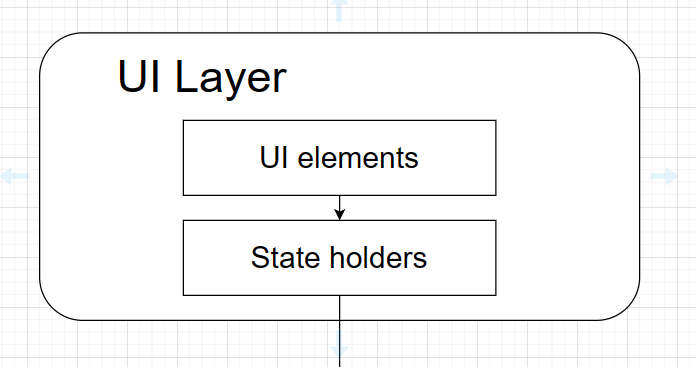
1. This project will use application architecture to divide the app into three parts, they are: User interface Layer, Domain Layer and Data Layer. And the architecture of applications is shown in Figure 1:



(Figure 1 Application architecture.)

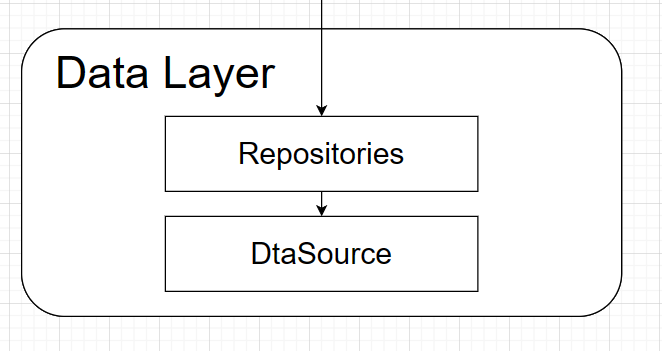
1. User interface Layer: The function of the User interface layer is to display application data on the screen. Whenever data changes, whether due to user interaction or external input, the interface should be updated to reflect these changes.

And there are two parts included in the User interface. And the architecture of User interface is shown in Figure 2:



(Figure 2: Components of UI Layer.)

1. Data Layer: The data layer consists of multiple warehouses, each of which can contain zero to multiple data sources. And the architecture of data layer is shown in Figure 3.



(Figure 3: Data Layer.)

1. Domain Layer: The Domain Layer is an optional layer between the interface and data layers. The Domain Layer is responsible for encapsulating complex business logic or simple business logic that is reused by multiple ViewModels.

Requirement Gathering Methods:

* + - 1. User research: Collect user needs and expectations for mobile software through questionnaire surveys, user interviews, and other methods to understand their dissatisfaction with existing software and the functions they hope to improve.
      2. Competitive analysis: Analyze similar mobile apps in the market, understand their functional characteristics and user experience, identify their advantages and disadvantages, and provide reference for software development.
      3. User experience analysis: Evaluate the user experience during the use of software through methods such as user research and prototype testing, including requirements for user-friendly interface and smooth interaction.

## Technology

1. Hardware specification:
   1. Processor: Intel Core 7
   2. Memory: 16GB
   3. Storage: 512GB SSD
   4. Graphics: NVIDIA GeForce RTX 2060
   5. Display: 15.6-inch Full HD screen
2. Software environment:
   1. Operating System: Windows 11
   2. Development Tool: Android Studio
   3. Programming Language: java, Kotlin
   4. Version Control: Git
   5. Design Tools: Adobe XD, Sketch

## Testing and Evaluation Plan

1. Unit Testing:
   1. Testing Plan: The unit testing plan involves identifying the specific units of code to be tested, creating test cases for each unit, and determining the expected outcomes. It also includes setting up the testing environment, such as using testing frameworks like JUnit or NUnit, and integrating the unit tests into the continuous integration pipeline.
   2. Evaluation Plan: The evaluation plan for unit testing involves running the unit tests regularly, ideally as part of the continuous integration process, and analyzing the test results. It includes checking for test failures, code coverage, and ensuring that the code meets the expected behavior.
   3. Test Case:

Unit: LoginService

Test Case: Verify that the login service correctly authenticates valid user credentials.

Expected Outcome: Successful authentication and access granted.

Failed Outcome: errorMessage.

1. Integration Testing:
   * + - 1. Testing Plan: The integration testing plan involves identifying the different components or modules to be tested, defining the integration points, and creating test cases to validate the interactions between these components. It also includes setting up the testing environment, such as using mock objects or stubs to simulate external dependencies.
         2. Evaluation Plan: The evaluation plan for integration testing involves running the integration tests to verify that the components work together as expected. It includes checking for successful integration, identifying any integration issues or failures, and ensuring that the overall system functionality meets the requirements.
         3. Test Case: Components: User Test Case: Test the flow of data from the User Interface (UI) to the database and back. Expected Outcome: Data is successfully stored and retrieved form the database. Failed Outcome: errorMessage.
2. Acceptance Testing:
   1. Testing Plan: The acceptance testing plan involves defining the acceptance criteria based on user requirements, creating test cases to validate the user interface, user experience, and overall functionality of the application. It also includes setting up the testing environment, such as using automated testing tools for UI testing.
   2. Evaluation Plan: The evaluation plan for acceptance testing involves running the acceptance tests to validate that the application meets the user requirements and provides a positive user experience. It includes gathering feedback from users, identifying any usability issues, and ensuring that the application meets the acceptance criteria.
   3. Test Case:

Feature: User Registration

Test Case: Verify that a new user can successfully register and receive a confirmation email.

Expected Outcome: User receives a confirmation email and can log in with the newly created account.

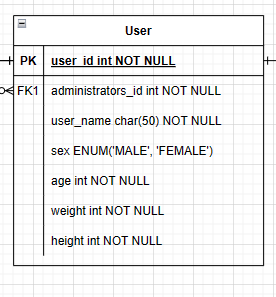
Failed Outcome: errorMessage.

These testing and evaluation plans are essential for ensuring the quality and reliability of the software developed using Test-driven development. They help in catching bugs early in the development process and ensuring that the software meets the requirements and expectations of the end users.

## Design and Implementation

This project was developed on the Android Studio development platform. And based on Kotlin language for code writing and logic implementation. So far, the project has been completed:

1. User Interface (UI) interface design
   1. A detailed page layout diagram was drawn based on the reference of existing software interfaces. Then write code based on the chart. And conducted a survey on whether friends and strangers around me liked the layout plan.
2. Database establishment
   1. A detailed E-R diagram was drawn. And establish a database based on the relationships in the E-R diagram. Here is an example from the E-R diagram:



(Figure 4 An example in the E-R diagram)

1. Login and registration code writing
   1. Successfully wrote login/registration page layout code. Write code and implement logic to achieve login/registration.

# Project Management

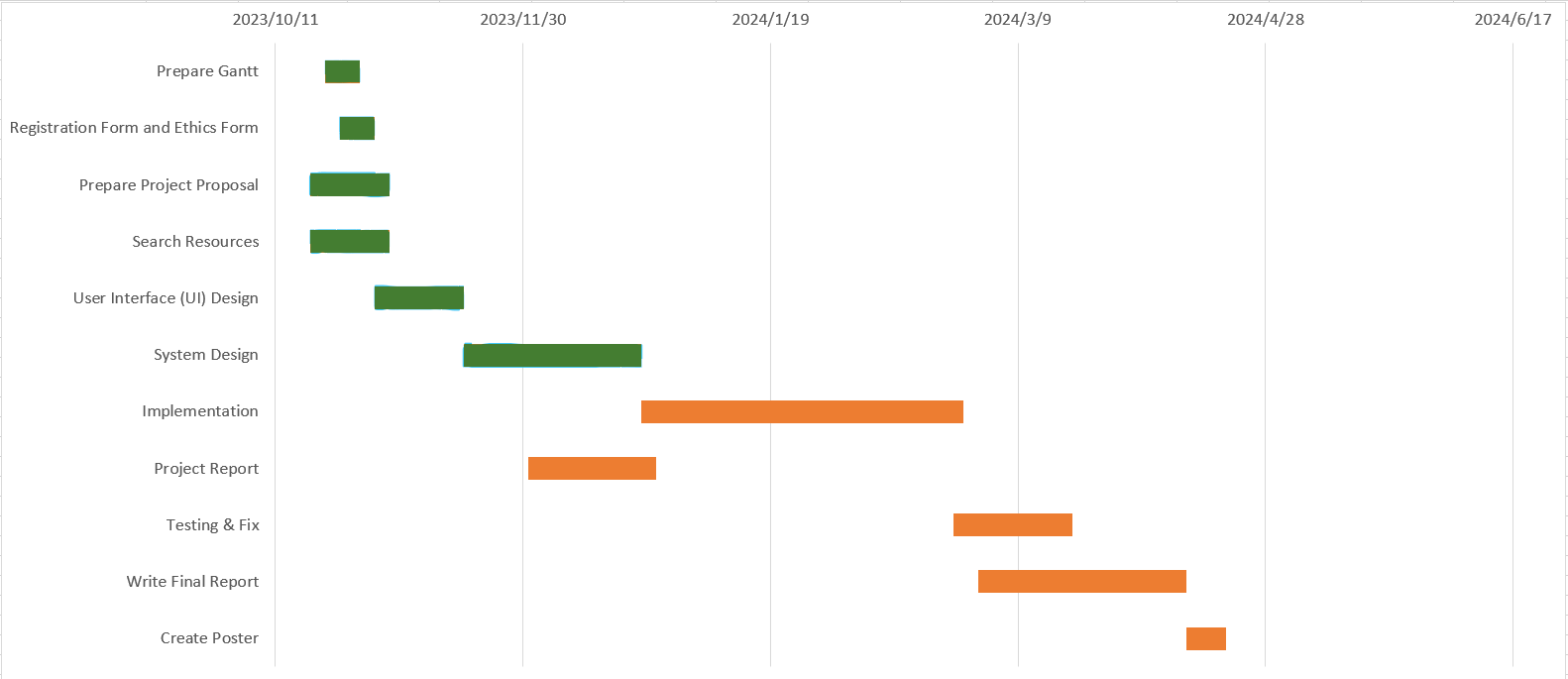
## Activities

|  |  |  |
| --- | --- | --- |
| Objectives | Complete | Uncomplete |
| Database | Create Database | Storage and retrieval of data |
| Interface | Register interface | Main interface |
| Login interface | User Personal interface |
|  | Search for recipes |
|  | Create a recipe |
| System |  | Generate shopping list |

Table 1 Complete and Uncomplete tasks

## Schedule

The green section represents tasks that have been completed or submitted for evaluation. The orange part represents tasks that have not been submitted for evaluation or are currently in progress.



(Figure 5 Gantt chart for project.)

## Project Version Management

After completing each section of the code. Directly upload to Baidu Cloud. And there is a detailed classification of each part of the code.



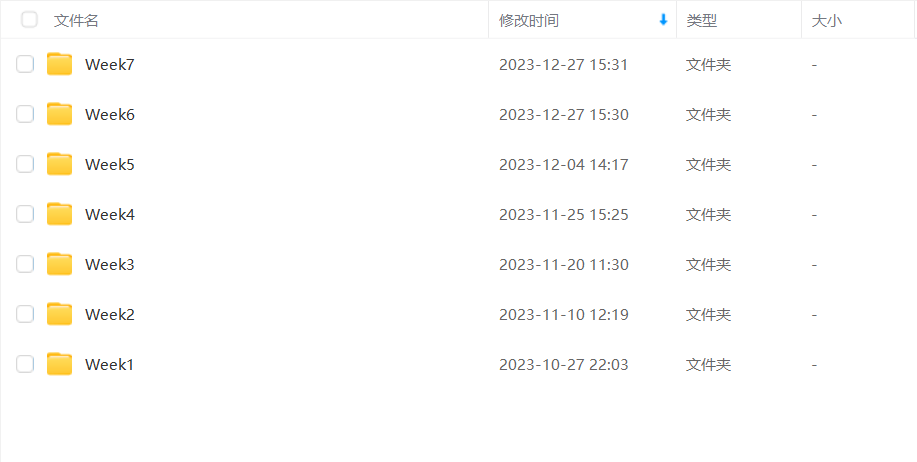
(Figure 6 The current code in Baidu Cloud.)

## Project Data Management

Store all found literature and submitted evaluation reports on Baidu Cloud. Including weekly reports submitted. And classify as shown in the following figure:



(Figure 7 The current resources in Baidu Cloud.)



(Figure 8 Weekly report of each week)

## Project Deliverables

Documents submitted for evaluation:

1. Project proposal
2. Weekly Report

Documents that have not yet been submitted for evaluation:

1. Project progress report.
2. Final report.
3. Project code / Software.
4. Presentation PPT

# Professional Issues and Risk:

## Risk Analysis

1. Risk analysis as informed by current progress:
   1. The current project is still in the development stage. During development, you may encounter many bugs and misses
   2. Current technical obstacles: learning a new language to write code, and falling behind schedule due to DDL
2. Resolved risks and the success of the mitigation strategy:
   1. Resolved risk: When working with the code, frequent errors occurred due to insufficient proficiency in the new programming language. The mitigation strategy involved continued learning and seeking assistance from friends with experience in similar projects, ultimately leading to the successful resolution of the errors.
   2. Analysis of impact and effectiveness of the strategy: The errors and insufficient proficiency have been a factor causing delays in the project progress. However, seeking help and continuing to learn the new language can help me grasp it more quickly. This can enable me to accelerate the progress in the future.
3. Possible risks in the future:
   1. Possible risks in the future: Software vulnerabilities and technical issues that cannot be solved by the knowledge currently learned.
   2. Mitigation strategy: Continuously learning knowledge or asking classmates for similar projects to solve technical problems. Conduct rigorous code testing, promptly fix vulnerabilities and issues, and maintain software stability and security.

## Professional Issues

When developing a mobile software project, it is important to consider the legal, social, ethical, and environmental issues that may arise in the context of the project. This includes adhering to professional codes of conduct, such as those set forth by organizations like the British Computer Society (BCS) and the Association for Computing Machinery (ACM).

Legal issues may include ensuring that the software complies with data protection laws, intellectual property rights, and any other relevant legislation. It is important to conduct a thorough legal review of the project to identify and address any potential legal risks.

Social issues may involve considering the impact of the software on society, including issues related to accessibility, diversity, and inclusion. It is important to ensure that the software is designed and developed in a way that is inclusive and accessible to all users.

Ethical considerations may include ensuring that the software is developed and used in a way that is fair, transparent, and respects the rights and dignity of individuals. This may involve considering the ethical implications of the data collected and how it is used, as well as ensuring that the software does not perpetuate biases or discrimination.

Environmental concerns may include taking into account the software's environmental impact, such as its energy usage and carbon footprint. It is critical to design and build software that has a low environmental effect.

In addition to considering these issues, it is important to refer to professional codes of conduct, such as those set forth by the BCS and ACM, to ensure that the project is developed and implemented in a professional and ethical manner. This may involve adhering to principles such as honesty, integrity, and respect for the rights of others.

By identifying and discussing these issues in the context of the project, developers can ensure that their mobile software project is developed and implemented in a way that is legal, socially responsible, ethical, and environmentally sustainable.

# References

[1] H. Cena and P. C. Calder, ‘Defining a Healthy Diet: Evidence for the Role of Contemporary Dietary Patterns in Health and Disease’, *Nutrients*, vol. 12, no. 2, Art. no. 2, Feb. 2020, doi: 10.3390/nu12020334.

[2] M. Amiri, J. Li, and W. Hasan, ‘Personalized Flexible Meal Planning for Individuals With Diet-Related Health Concerns: System Design and Feasibility Validation Study’, *JMIR formative research*, vol. 7, p. e46434, Aug. 2023, doi: 10.2196/46434.

[3] H. Punamiya, S. Mavani, M. Jain, and A. Dhruv, ‘Food recipe application’, Accessed: Oct. 27, 2023. [Online]. Available: https://www.academia.edu/37567801/Food\_recipe\_application

[4] A. Sharma, A. Karamchandani, D. Dave, A. Patel, and N. Doshi, ‘Database Management Systems—An Efficient, Effective, and Augmented Approach for Organizations’, in *ICT with Intelligent Applications*, T. Senjyu, P. N. Mahalle, T. Perumal, and A. Joshi, Eds., in Smart Innovation, Systems and Technologies. Singapore: Springer, 2022, pp. 465–478. doi: 10.1007/978-981-16-4177-0\_47.

[5] YonBuilder. Accessed: Oct. 27, 2023. [Online]. Available: <https://developer.yonyou.com/product/appdevelopment/appdriven>

[6] Developers. Accessed: Oct. 27, 2023. [Online]. Available: <https://developer.android.google.cn/?hl=zh-cn>

[7] Paprika Recipe Manager. Accessed: Dec. 27, 2023. [Online]. Available: <https://www.paprikaapp.com/>