**ASSIGNMENT 1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Qualification** | **BTEC Level 5 HND Diploma in Computing** | | | | |
| **Unit number and title** | **Unit 22: Application Development** | | | | |
| **Submission date** |  | | | **Date Received 1st submission** |  |
| **Re-submission Date** |  | | | **Date Received 2nd submission** |  |
| **Group number:** |  | **Student names & codes** |  | **Final scores** | **Signatures** |
| 1 | Tran Cong Hoang | BH00317 |  |  |
| 2 | Nguyen Viet Anh | BH00404 |  |  |
| 3 | Ngo Vi Thien Quang | BH00589 |  |  |
| 4 | Dinh Hoang Duong | BH00631 |  |  |
| **Class** |  | | | **Assessor name** |  |
| **Student declaration**  I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice. | | | | | |

**Grading grid**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **P1** | **P2** | **M1** | **P3** | **M2** | **D1** |
|  |  |  |  |  |  |

**OBSERVATION RECORD**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student 1** |  | | |
| **Description of activity undertaken**  **• What the learner did**  **• The evidence provided/questions asked, and answers given:** | | | |
|  | | | |
| **Assessment & grading criteria** | | | |
|  | | | |
| **How the activity meets the requirements of the criteria** | | | |
|  | | | |
| **Student signature:** |  | **Date:** |  |
| **Assessor signature:** |  | **Date:** |  |
| **Assessor name:** |  | | |
| **Student 2** |  | | |
| **Description of activity undertaken** | | | |
|  | | | |
| **Assessment & grading criteria** | | | |
|  | | | |
| **How the activity meets the requirements of the criteria** | | | |
|  | | | |
| **Student signature:** |  | **Date:** |  |
| **Assessor signature:** |  | **Date:** |  |
| **Assessor name:** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Student 3** |  | | |
| **Description of activity undertaken** | | | |
|  | | | |
| **Assessment & grading criteria** | | | |
|  | | | |
| **How the activity meets the requirements of the criteria** | | | |
|  | | | |
| **Student signature:** |  | **Date:** |  |
| **Assessor signature:** |  | **Date:** |  |
| **Assessor name:** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Student 4** |  | | |
| **Description of activity undertaken** | | | |
|  | | | |
| **Assessment & grading criteria** | | | |
|  | | | |
| **How the activity meets the requirements of the criteria** | | | |
|  | | | |
| **Student signature:** |  | **Date:** |  |
| **Assessor signature:** |  | **Date:** |  |
| **Assessor name:** |  | | |

|  |  |  |
| --- | --- | --- |
| **Summative Feedback:  Resubmission Feedback:** | | |
| **Grade:** | **Assessor Signature:** | **Date:** |
| **Internal Verifier’s Comments:** | | |
| **Signature & Date:** | | |

|  |
| --- |
| Table of Contents  [**I.** **Introduction** 3](#_Toc151380541)  [**II.** **Contents.** 5](#_Toc151380542)  [**Activity 2: Appropriate software development tools and techniques to deploy the solution.** 5](#_Toc151380543)  [**1)** **The strengths and weaknesses of SDLC models can be applied to develop the "CampusExpense Manager" application.** 5](#_Toc151380544)  [**a)** **Agile models** 5](#_Toc151380545)  [**b)** **Waterfall model** 6](#_Toc151380546)  [**c)** **Spiral model.** 7](#_Toc151380547)  [**d)** **V model.** 8](#_Toc151380548)  [**e)** **RAD models.** 9](#_Toc151380549)  [**2)** **Development tools for project implementation.** 10](#_Toc151380550)  [**a)** **Project management tools** 10](#_Toc151380551)  [**b)** **Design tool** 10](#_Toc151380552)  [**c)** **IDE** 11](#_Toc151380553)  [**3)** **Justifying development tools and development methodology selected.** 11](#_Toc151380554)  [**a)** **Choosing SDLC model** 11](#_Toc151380555)  [**b)** **Choosing developing tools** 12](#_Toc151380556)  [**Activity 3: Analyze business problems to come up with proposed solutions.** 15](#_Toc151380557)  [**1)** **Software design documents for the Android application "CampusExpense Manager".** 15](#_Toc151380558)  [**A.** **UML diagram** 15](#_Toc151380559)  [**B.** **Software development documents** 17](#_Toc151380560)  [**2)** **Create a test plan for "CampusExpense Manager".** 1](#_Toc151380561)  [**3)** **Why would this design work well for the "CampusExpense Manager" project?** 1](#_Toc151380562)  [**Activity 4: Evaluate your group's solution by comparing it with others.** 3](#_Toc151380563)  [**1)** **The SDLC model is chosen over other SDLC models.** 3](#_Toc151380564)  [**2)** **Selected tools compared to other development tools.** 4](#_Toc151380565)  [**III.** **Conclusion** 6](#_Toc151380566)  [**IV.** **References** 7](#_Toc151380567) |

|  |
| --- |
| List of Tables  [Table 1: Resource and financial cost 19](#_Toc151380568)  [Table 2: Test case functional Requirements 6](#_Toc151380569)  [Table 3:Test case non-functional Requirements 10](#_Toc151380570) |

|  |
| --- |
| List of Figures  [Figure 1: Agile models 5](file:///C:\BTEC_FPT\IT0601-Application%20Development\TranCongHoang_BH00317_ASM1_2nd.docx#_Toc151380571)  [Figure 2: Waterfall model 6](file:///C:\BTEC_FPT\IT0601-Application%20Development\TranCongHoang_BH00317_ASM1_2nd.docx#_Toc151380572)  [Figure 3: Spiral model 7](file:///C:\BTEC_FPT\IT0601-Application%20Development\TranCongHoang_BH00317_ASM1_2nd.docx#_Toc151380573)  [Figure 4: V model 8](file:///C:\BTEC_FPT\IT0601-Application%20Development\TranCongHoang_BH00317_ASM1_2nd.docx#_Toc151380574)  [Figure 5: RAD model 9](file:///C:\BTEC_FPT\IT0601-Application%20Development\TranCongHoang_BH00317_ASM1_2nd.docx#_Toc151380575)  [Figure 6: Class diagram 15](file:///C:\BTEC_FPT\IT0601-Application%20Development\TranCongHoang_BH00317_ASM1_2nd.docx#_Toc151380576)  [Figure 7: Use case diagram 16](file:///C:\BTEC_FPT\IT0601-Application%20Development\TranCongHoang_BH00317_ASM1_2nd.docx#_Toc151380577)  [Figure 8: Work breakdown structure 17](file:///C:\BTEC_FPT\IT0601-Application%20Development\TranCongHoang_BH00317_ASM1_2nd.docx#_Toc151380578)  [Figure 10: Timeline and milestones 18](file:///C:\BTEC_FPT\IT0601-Application%20Development\TranCongHoang_BH00317_ASM1_2nd.docx#_Toc151380579)  [Figure 9: Timeline and milestones 18](file:///C:\BTEC_FPT\IT0601-Application%20Development\TranCongHoang_BH00317_ASM1_2nd.docx#_Toc151380580)  [Figure 11: Timeline and milestones 19](file:///C:\BTEC_FPT\IT0601-Application%20Development\TranCongHoang_BH00317_ASM1_2nd.docx#_Toc151380581) |

# **Introduction**

Students are a group of people with high needs for spending and financial management. They face many expenses related to study, living, entertainment, etc. They must also seek out and apply for student incentives for the products and services they use. Furthermore, they also have to manage financial grants such as scholarships, student loans, part-time jobs, etc. Additionally, they also need to access and use school resources such as the library, gym, health center, etc. All of these factors require students to have a tool that helps them track spending, and budget, and save time.

The purpose of this report is to introduce the mobile application "CampusExpense Manager", a tool that helps students manage to spend effectively and save time. This app is developed by BudgetWise Solutions, a small development team with limited experience in mobile application development. This application is designed specifically for students, with main features such as:

* Expenditure tracking: The app allows users to record income and expenses in different categories, such as food, rent, utilities, entertainment, etc. Users can also attach receipts, notes, or photos to each transaction.
* Budgeting: The app helps users set up monthly or weekly budgets for different spending categories. The app also shows how much the user has spent and how much is left. Users can also view their spending trends and habits over time.
* Graphs and charts: The app generates reports and charts to visualize users' income and spending by different categories, time periods,
* or locations. Users can also export their data to CSV or PDF files for backup or analysis.
* Notifications and reminders: The app sends notifications and reminders to users when they are close to or over their budget limits for specific categories, when they have bills or payments coming up, or when they reach their savings goal.

This report will limit its scope to the main features of the application and will not go into technical or design details. We will not compare this application with other spending management applications on the market, but will only analyze the needs and spending habits of students, as well as the financial problems they often encounter. We also will not discuss legal, ethical or social issues related to the use of this application.

To develop this application, we used appropriate software development tools and techniques to implement the solution. We chose the Agile SDLC model to manage the software development process, because it allows us to adapt quickly to changing customer requirements, enhance collaboration among team members, and customers, improve product quality by continuous testing and receive early feedback. We also used the Xamarin cross-platform development tool to program the app for both Android and iOS, as it saves development costs and time by using a single source code, taking advantage of the the latest and most popular technology in mobile programming, creating apps with near-native look and performance.

To come up with proposed solutions to business problems, we analyzed students' needs and spending habits, as well as the financial problems they often face. We designed the application's user interface and business logic according to modern design principles, such as user-centered design, functional design, and design. goal-oriented design. We also tested the quality and performance of the application using different types of testing, like white box, black box, gray box testing; Test functionality, performance, security, usability, compatibility, etc.

The "CampusExpense Manager" application has the following novelties and contributions to users and the community:

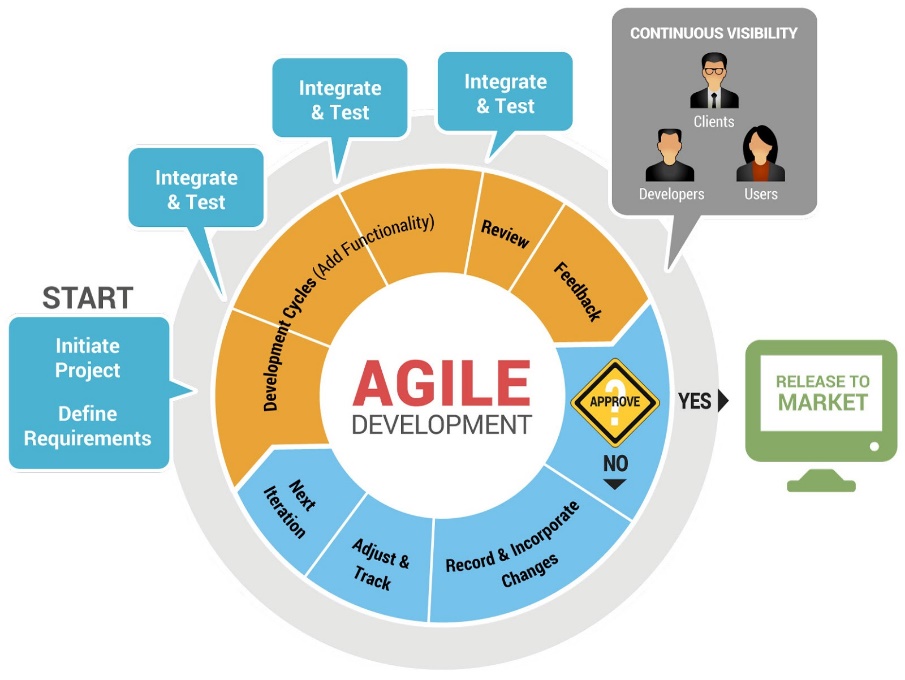
* Provides a friendly and easy-to-use interface, helping users manage spending quickly and conveniently.
* Able to operate on many mobile platforms such as Android and iOS, allowing users to use the application on any device they have.
* Has a feature to search and apply student incentives for products and services that users use, helping users save costs and increase value for their money.
* Has a feature to manage financial grants such as scholarships, student loans, part-time jobs, etc., helping users monitor their financial situation and plan to repay debt or save appropriately.
* Features access and use of school resources such as libraries, gym, medical center, etc., helping users enjoy school facilities and services without wasting time or money .

# **Contents.**

## **Activity 2: Appropriate software development tools and techniques to deploy the solution.**

### **The strengths and weaknesses of SDLC models can be applied to develop the "CampusExpense Manager" application.**

#### **Agile models**

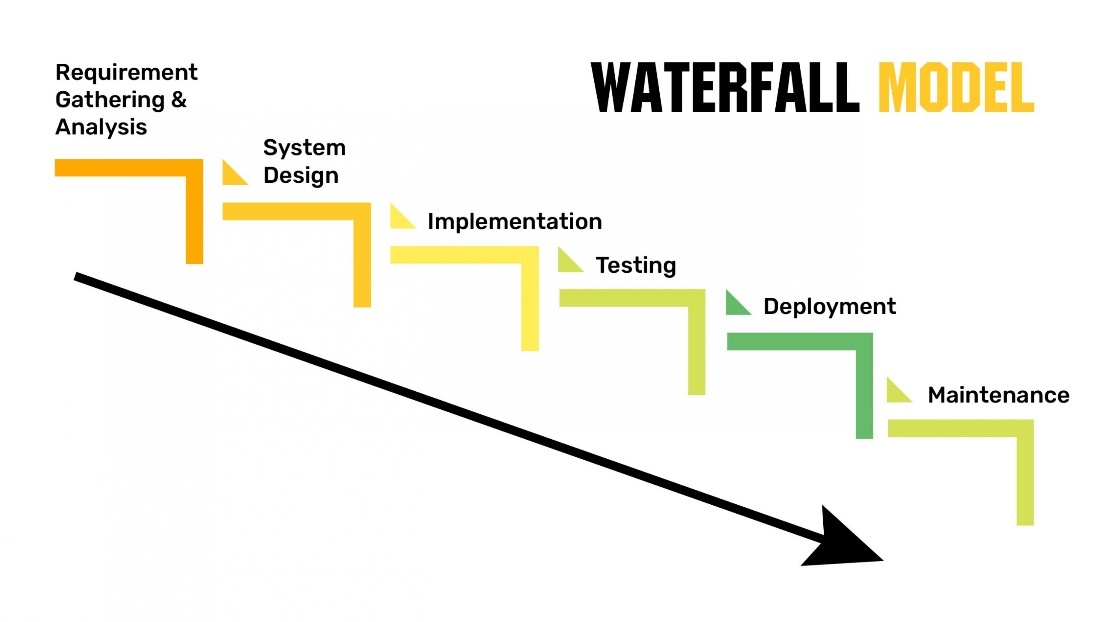
Agile software development is a software development methodology that values flexibility, collaboration, and customer satisfactio. Agile Software Development is an iterative and incremental approach to software development that emphasizes the importance of delivering a working product quickly and frequently. It involves close collaboration between the development team and the customer to ensure that the product meets their needs and expectations (Naidu, 2018).

* Strengths for the project "CampusExpense Manager".
* It is possible to create an application that matches the needs and expectations of users, by shipping working versions of the application in a short time and receiving feedback from them.
* Able to improve and adapt your application to changing requirements and environments, using advanced tools and techniques, such as code reuse, user interface development, functional component development, etc.

Figure : Agile models

* Able to work effectively in teams and communicate well with team members, customers and stakeholders, using methods of collaboration and information exchange, such as daily meetings, job boards, chat tools, and more.
* Can reduce the risk and cost of your project, by identifying and resolving issues early in the development process, such as application bugs or defects.
* Weaknesses for the project "CampusExpense Manager".
* It can be difficult to ensure the commitment and participation of users and stakeholders, as they may not always be available or feasible to provide feedback to your application.
* May not be able to meet the complex or large-scale requirements of your application, where you need to comply with data security or platform compatibility standards or regulations.
* May be incompatible with some organizational cultures or structures that prefer more formal and rigid processes, where you need to follow university policies or regulations.
* may depend on the skills and experience of team members, as they must make decisions and solve problems quickly and independently.

#### **Waterfall model**

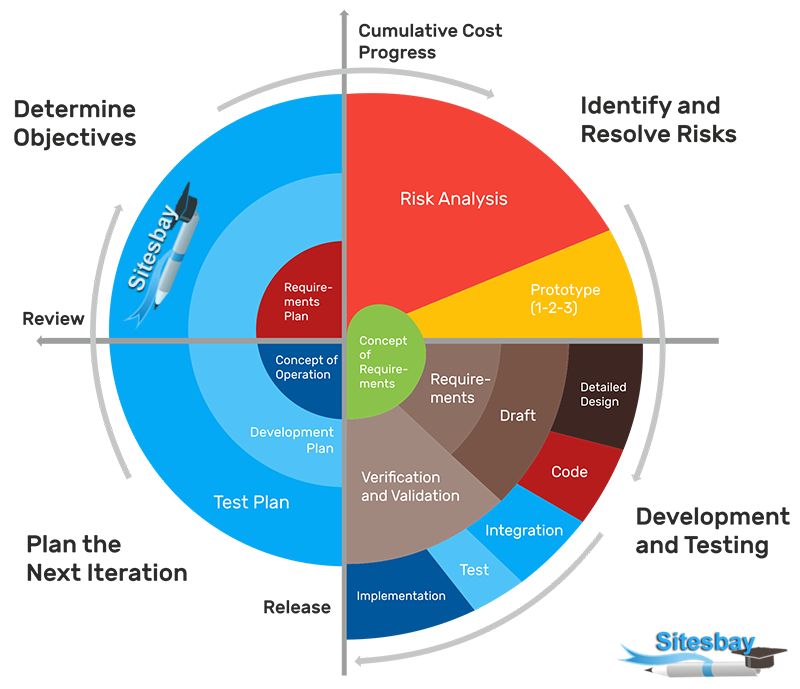
The waterfall model is a software development process model that follows a linear and sequential approach. It consists of five main phases: requirements analysis and specification, design, implementation and unit testing, integration and system testing, and operation and maintenance. The waterfall model is one of the earliest and simplest models to be used in software engineering. It is based on the idea that each phase must be completed and verified before the next phase can begin, and there is no overlap or iteration between the phases.

* Strengths for the project "CampusExpense Manager".
* Able to create a high quality and reliable application, by following a clear and structured framework for the development process, with requirements, specifications, deliverables and clearly defined milestones.

Figure : Waterfall model

* Able to document and verify each stage of the development process, using testing and verification methods, such as unit testing, integration testing, system testing, etc.
* Able to better control and manage project scope, budget, schedule, and resources, using project management tools and techniques, such as Microsoft Project, Gantt Chart, etc.
* May be suitable for projects with clear and fixed requirements and expectations from the start, such as platform compatibility or monetization strategy.
* Weaknesses of the project "CampusExpense Manager".
* May not be able to meet user needs and expectations, as they may have changes or feedback during the development of your application.
* Can increase risk and cost to your project, when you delay testing and integration until the end of development, which can lead to the late discovery of bugs or defects application.
* This can inhibit collaboration and communication between developers, customers, and stakeholders, as they have limited interaction and participation in the project.

#### **Spiral model.**

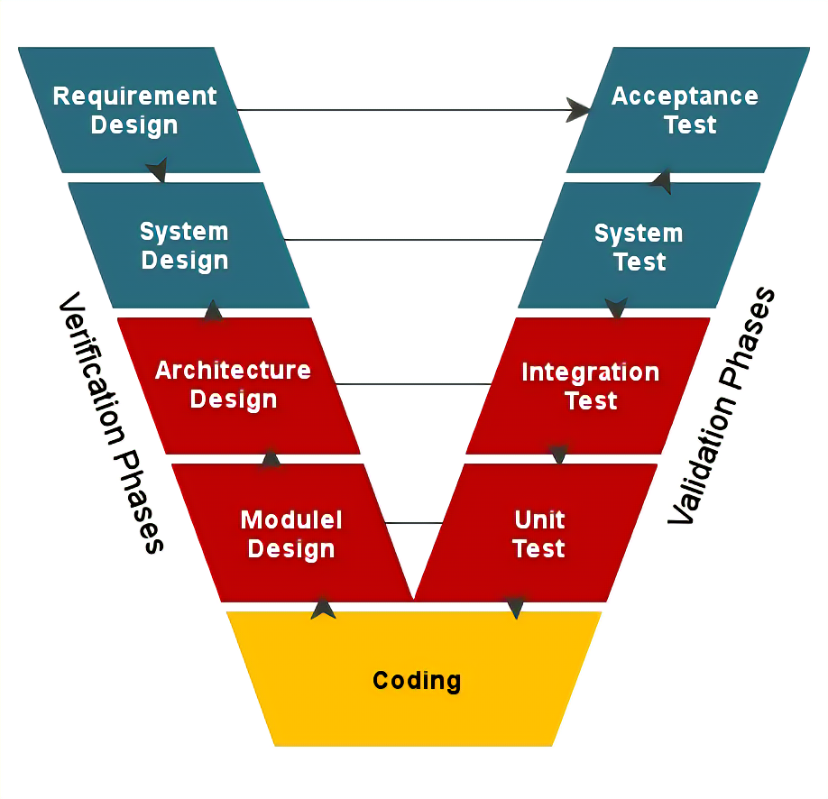
The Spiral Model is a software development life cycle model that provides support for risk handling. It is one of the most important SDLC models and is based on the idea of a spiral with many loops. Each loop of the spiral is called a phase of the software development process (Anon, 2022). The exact number of phases needed to develop the product can be varied by the project manager depending upon the project risks (Pal, 2018).

* Strengths for the project "CampusExpense Manager".
  + Able to create an application that is flexible and adaptable to changing requirements and environments, by modifying or improving your application at each iteration based on feedback and reviews from users and related parties.
  + Can reduce the risk and cost of your project, by identifying and resolving issues early in the development process, and conducting risk analysis at each iteration.
  + Can ensure the quality and reliability of your application, by applying testing and verification methods at every iteration.

Figure : Spiral model

* + May be suitable for projects with uncertain or dynamic requirements or expectations, such as tracking spending or setting budgets.
* Weaknesses of the project "CampusExpense Manager".
* It can be difficult to estimate project scope, budget, timetable, and resources, as they may change depending on each iteration.
* Multiple documents and reports may be involved for each iteration, which can add cost and complexity to your project.
* May depend on the presence and cooperation of users and stakeholders, as they must provide feedback and evaluation at each iteration.

#### **V model.**

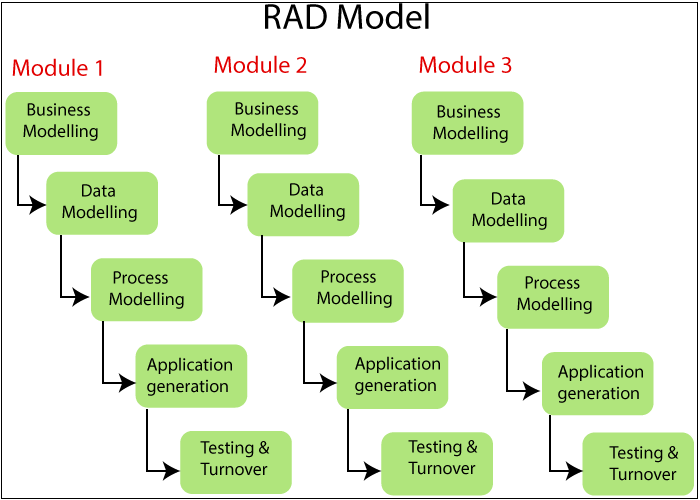
The V-model is a software development process that describes the relationship between each phase of the development life cycle and its corresponding testing phase (Anon, 2023). It is used to produce rigorous development lifecycle models and project management models. In this model, each phase of SDLC must be completed before the next phase starts. It follows a sequential design process same as the waterfall model (Rajkumar, 2019).

* Strengths for the project "CampusExpense Manager".
* It is possible to create a high-quality and reliable application, by following a clear and structured framework for the development process, with requirements, specifications, deliverables, and Milestones clearly defined.
* Each stage of the development process can be documented and verified, using testing and verification methods such as unit testing, integration testing, system testing, etc.
* Able to better control and manage project scope, budget, schedule, and resources, using project management tools and techniques, such as Microsoft Project, Gantt Chart, etc.
* May be suitable for projects with clear and fixed requirements and expectations from the start, such as platform compatibility or monetization strategy.

Figure : V model

* Weaknesses for the project "CampusExpense Manager".
* Users' needs and expectations may not be met, as they may have changes or feedback during the development of your application.
* Can increase risk and cost to your project, when you delay testing and integration until the end of development, which can lead to the late discovery of bugs or defects application.
* Collaboration and communication between developers, customers, and stakeholders can be inhibited, as they have limited interaction and participation in the project.

#### **RAD models.**

The RAD model is a software development approach that emphasizes rapid prototyping and feedback over detailed planning and documentation. It is suitable for projects that have changing requirements, user involvement, and short delivery time. The RAD model has four main phases: business modeling, data modeling, process modeling, and application generation.

* Strengths for the project "CampusExpense Manager".
* It is possible to create an application quickly and meet user needs, using advanced tools and techniques to develop applications in a short time.
* Able to engage and receive feedback from users during the development of your application, using requirements gathering, user design, user interface building, component building methods function, etc.

Figure : RAD model

* Can use advanced tools and techniques to increase the efficiency and quality of your application, such as code reuse, user interface development, functional component development, etc.
* May be suitable for projects with dynamic or unclear requirements, such as tracking spending or setting budgets.
* Weaknesses of the project "CampusExpense Manager".
* It can be difficult to ensure user commitment and participation, as they may not always be available or feasible to provide feedback to your app.
* May not be able to meet the complex or large-scale requirements of your application, where you need to comply with data security or platform compatibility standards or regulations.
* May not be compatible with some organizational cultures or structures that prefer more formal and rigid processes, where you need to follow university policies or regulations.
* May depend on the availability and quality of the tools and techniques used, which may affect application performance and reliability.

### **Development tools for project implementation.**

#### **Project management tools**

Project management tools help create, assign, and track tasks, and project schedules, and manage and allocate resources effectively. It also provides features for team collaboration and communication, to enhance communication, encourage collaboration, and keep all team members updated. Examples of project management tools are:

* + **Microsoft Project**: This is a project management software that helps us plan, track, and manage project activities. It allows us to create tasks, assign resources, set deadlines, monitor progress, generate reports, and more.
  + **GitHub**: This is an online source code hosting service that supports collaboration between team members and provides version control. It enables effective source code management, change tracking, and problem resolution.
  + **Gantt Project**: This is a Gantt chart tool, with the ability to visualize project schedules. It helps outline and communicate the development plan for the CampusExpense Manager project. It helps team members and stakeholders clearly understand milestones and deadlines.

#### **Design tool**

The design tool is used to create visual designs and prototypes for applications. It includes design for user interfaces, wireframes, animations, graphic design, etc. Examples of design tools are:

* + **Canva**: This is an online graphic design tool, used to create attractive user interfaces and graphics for the CampusExpense Manager application. It allows the design team to create visual layouts, icons, and other graphic elements suitable for college students.
  + **Draw.io**: This is an online diagramming tool that plays an important role in creating interactive prototypes and flow diagrams for application functionality. It helps map user flows and interaction sequences, ensuring that the app's features align with the financial management needs of college students.
  + **UML diagrams**: These are visual models that represent different aspects of our software system, like classes, objects, components, interactions, etc. They help us analyze, design, and document our software systems using a standard notation.

#### **IDE**

IDE is a software application that provides a comprehensive environment for software development. It includes features like a code editor, debugger, build tools, and version control integration. For example: Visual Studio code, Android studio, IntelliJ IDEA.

* **Android Studio**: This is the main development environment for the CampusExpense Manager project, specifically for the Android version of the application. It provides specialized tools for Android application development, simplifying the design, programming, and testing processes.
* **Visual Studio Code**: This is a popular source code editor, developed with wide language support, lightweight, cross-platform, and has an IDE, great for developing small projects and many devices such as the CampusExpense Manager project.
* **IntelliJ IDEA**: IntelliJ IDEA is an IDE (integrated development environment) for Java and Kotlin developers. It provides many features to support programming, such as code editor, debugger, build tools, version control integration, and more. It also supports many web and enterprise frameworks and platforms, such as Spring, Jakarta EE, Hibernate, Android, etc. In addition, it has other integrated development tools, such as database tools and performance analysis, that allow you to manage and optimize your application effectively.

### **Justifying development tools and development methodology selected.**

#### **Choosing SDLC model**

After careful consideration, our team chose the Rapid Application Development (RAD) model as the most suitable method for the CampusExpense Manager project. This decision is in accordance with the special needs and constraints of the project, and the following reasons are the basis for this choice:

* + **Speed and Agility**: The RAD model is known for its ability to develop quickly, which is an important factor in achieving project completion times. By emphasizing speed and agility, RAD allows us to react quickly to changing requirements and deliver a working product within set timescales.
  + **Flexibility and Adaptation**: As the CampusExpense Manager project progresses, we anticipate the need for flexibility and adaptation. User feedback and changing financial management needs require a development approach that can adapt to these adjustments. The iterative nature of RAD makes it suitable for handling changing user requirements and market dynamics. For example, users may want to add some new features or remove some unnecessary features to the application, or the market may have new trends in financial management for students.
  + **Efficient use of resources**: Given the constraints of a limited budget and the average experience of the development team, RAD provides an efficient resource management approach. It allows us to break down the development process into small, manageable components that match the skills and abilities of team members. This approach enhances their learning and productivity throughout the project.
  + **Active user participation**: Incorporating active user participation is a key element of RAD. This aligns perfectly with our goal of creating a user-focused CampusExpense Manager application. By participating in the development process, we ensure that the app directly meets the financial management needs of students. Their ongoing feedback will guide feature adjustments and additions.
  + **Risk management**: The loops of the RAD model also serve as a risk mitigation strategy. In a project with a strict deadline, detecting and resolving potential problems early is critical. RAD's short development iterations allow us to identify and resolve challenges promptly, minimizing the likelihood of major project problems. For example, we can detect and fix bugs, code conflicts, or performance issues in each iteration, before they become serious problems that affect the entire application.
  + **High-quality results**: Ensuring a high-quality user experience is important. RAD's approach incorporates testing and quality assurance throughout each iteration, driving the delivery of a CampusExpense Manager application that is reliable and easy to use, even when processing large amounts of spending data.
  + **Stakeholder engagement**: The RAD model encourages active participation of stakeholders, including university administrators and students. This approach ensures that their views are continuously considered throughout the development process. In the case of university administrators, their concerns about promoting financial management tools can be directly addressed through regular interactions in RAD loops.

#### **Choosing developing tools**

##### GitHub

GitHub is an online source code hosting service that helps manage and collaborate with development team members. We chose GitHub because it has the following benefits:

* + It allows tracking and controlling versions of the source code, ensuring that you always have the latest updates and can roll back previous changes if needed.
  + It allows creating branches to develop separate features without affecting the main source code (master).
  + It allows creating pull requests to send and receive feedback on your source code from other team members and merge changes into the main source code once approved.
  + It allows creating issues to report and resolve bugs, problems, or new requests related to your project.
  + It allows the use of external tools and services, such as Firebase, MongoDB, Instabug, etc., to integrate into your project through application programming interfaces (APIs).

##### Microsoft Project

Microsoft Project is a project management software that helps plan, track, and manage project activities. We use Microsoft Project because it has the following benefits:

* + It allows for creating tasks, assigning resources, setting deadlines, establishing relationships and dependencies between tasks, and defining project milestones.
  + It allows for tracking project progress, costs, time,and quality, and comparing with the original plan to identify deviations and risks.
  + It allows you to create detailed and intuitive reports on projects, including Gantt charts, progress charts, resource charts, cost charts, and more.
  + It enables collaboration and communication with team members and other project stakeholders, through features like email, messages, calendar, etc.

##### Android Studio

Android Studio is an integrated development environment (IDE) for Android application development. We chose Android Studio because it has the following benefits:

* + It supports many programming languages, such as Java, Kotlin, and C++, and provides features such as code editor, debugger, build tools, etc.
  + It provides specialized tools for Android app development, such as interface design, emulator, testing tools, etc.
  + It provides a rich set of libraries and components to create interfaces and functions for Android applications, such as Material Design, Jetpack, Firebase, etc.
  + It provides a platform to distribute and publish your Android application on the Google Play Store.

##### Canva

Canva is an online graphic design tool that helps you create attractive interfaces and graphics for the CampusExpense Manager application. We use Canva because it has the following benefits:

* + It provides many design templates and customization features for the application interface, reducing the time needed to design a new template for the CampusExpense Manager application.
  + It allows the creation of visual layouts, icons, and other graphic elements suitable for college students, the target users of the application.
  + It provides collaboration and sharing features, making it easy for team members to share their design work and receive feedback.
  + It is an online and free tool, which means team members do not have to download additional software to design and save on the project budget.

##### draw.io

draw.io is an online diagramming tool that creates UML diagrams for CampusExpense Manager documents, such as use case diagrams, class diagrams, and database diagrams. We chose draw.io because it has the following benefits:

* + It has many tools, templates, and symbols for drawing different types of diagrams, and has a PDF export feature for storing and sharing diagrams.
  + It helps to represent the elements, relationships, and behaviors of the CampusExpense Manager system, and communicate effectively with different stakeholders, such as users, administrators, and developers.
  + It is an online and free tool, which means team members do not have to download additional software to draw diagrams and save on the project budget.

## **Activity 3: Analyze business problems to come up with proposed solutions.**

### **Software design documents for the Android application "CampusExpense Manager".**

#### **UML diagram**

##### Không có mô tả.Class diagram

Figure : Class diagram

Class "User": This table stores information about the user, including username, password, email and id. This table has a one-to-many relationship with the Month expense, Recurring expense and Budget categories tables, meaning each user can have multiple records in these tables.

Class "Month expense": This table stores information about the user's monthly expenses, including description, date, amount and category. This table has a many-to-many relationship with the Budget categories table, meaning each expenditure can belong to many categories, and each category can have many expenditures.

Class "Recurring expenses": This table stores information about users' recurring expenses, including description, start date, end date, amount and category. This table also has a many-to-many relationship with the Budget categories table, similar to the Month expense table.

Class "Budget categories": This table stores information about the user's budget categories, including name and budget. This table has a many-to-many relationship with the Month expense and Recurring expense tables, as mentioned above.

##### A diagram of a company Description automatically generatedUse case diagram

Figure : Use case diagram

Your use case includes two types of actors: First Actor and Second Actor. The First Actor is the user, i.e. the student, who uses the main functionality of the application. The second actor is the administrator, who manages the application, and the user, who uses the application's secondary functions.

The use cases in my App are:

- Registration: Allows users to create a new account in the system by providing personal information and creating login information.

- Login: Allows users to authenticate themselves and access the application using registered credentials.

- Logout: Allows users to end the current session and securely log out of the application.

- Set up a budget: Allows users to define spending limits and budget goals for different spending categories.

- Spend tracking: Allows users to add, edit, categorize and manage their spending. This includes adding new spend, modifying existing spend, assigning categories to spend, and deleting spend if necessary.

- See spending overview: Allows users to see a summary of their monthly spending, including total amount spent, amount remaining in budget, and percentage spent by each category . Users can also view their spending trends over time using a chart or table.

- Get spending alerts: Allows users to receive notifications or reminders when they approach or exceed their budget limit for a specific category. Users can also customize their notification settings, such as frequency, sound, or turn off notifications.

- Feedback and support: Allows users to send feedback or report problems about the application to the development team. Users can also view frequently asked questions or contact support staff if needed.

- Create spending reports: Allows administrators to create reports for analytical and administrative purposes, such as aggregating spending across all users or specific user groups.

#### **Software development documents**

##### A diagram of a company Description automatically generatedWork breakdown structure

Figure : Work breakdown structure

##### A screenshot of a project Description automatically generatedA screenshot of a computer Description automatically generatedTimeline and milestones

Figure : Timeline and milestones

Figure : Timeline and milestones

A screenshot of a computer

Description automatically generated

Figure : Timeline and milestones

##### Resource and financial cost

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Name expense** | **Number Needed** | **Cost per 1** | **Total** |
| **1** | **BudgetWise Solutions Teams** | | | **6600$** |
| 1.1 | Research team | 1 | 200$/ employee | 600$ |
| 1.2 | Development Team | 4 | 400$/ employee | 4800$ |
| 1.3 | Assurance Team | 3 | 200$/ times | 600$ |
| 1.4 | Media Team | 1 | 100$/ employee | 300$ |
| 1.5 | Financial Team | 1 | 100$/ employee | 300$ |
| **2** | **Tools** | | | **800$** |
| 2.1 | Microsoft 365 | 4 x 5 | 10$/ months | 200$ |
| 2.2 | Media advertising | 3 | 100$/ months | 300$ |
| 2.3 | App market | 3 | 100$/ months | 300$ |
| **3** | **Backup budget** | | | **2400$** |
|  | **Total** | | | **10000$** |

Table : Resource and financial cost

##### Feasibility report

To understand if a project can be successful or not, we must determine if our project is feasible or not. This includes evaluating the project from multiple perspectives such as technical, financial, market, and organizational feasibility to see if the idea or the project can become viable or not.

###### Technical feasibility

This is the aspect that involves determining whether the project can be carried out using existing technology. Factors to consider include platform type, programming language, system integration, data security, and scalability. According to your requirements, the CampusExpense Manager application must be developed for both Android and iOS platforms, using artificial intelligence and machine learning to analyze user spending behavior and integrate with bank accounts. users' credit cards and accounts, and has offline functionality. These are quite high and complex technical requirements, requiring the development team to have experience and expertise in mobile programming, artificial intelligence, and data security. However, there are many tools and resources available to support mobile app development, such as frameworks like React Native or Flutter, platforms like Firebase or AWS, or APIs like Google Maps or Flutter. Stripe. Therefore, I assess the project's technical capabilities as feasible, but it is necessary to choose the appropriate technology and provide skills training for the development team.

###### Economic feasibility

Economic viability: This is an aspect related to determining whether the project will bring economic benefits to stakeholders. Factors to consider include development costs, operating costs, expected revenue, and profits. According to your constraints, the project has a limited budget for development and marketing. This may affect the quality and performance of the application. However, there are several ways to minimize development costs, such as using free or open-source tools, hiring remote workers, or adopting a lean startup approach. In addition, the CampusExpense Manager application has high-profit potential due to a large target market of university students, can attract the interest and support of partners and sponsors, and May apply monetization strategies such as in-app advertising or charging for premium features. Therefore, I assess the economic viability of the project as **feasible**, but careful financial planning and monitoring of investment efficiency are required.

###### Legal feasibility

This is the aspect involved in determining whether the project complies with relevant laws and regulations. Factors to consider include copyright, patents, licenses, contracts, and social responsibility. At your request, the CampusExpense Manager application must comply with user data privacy and security regulations. This can impact project development and compliance costs. However, there are several ways to ensure the legality of your app, such as using encryption and data protection solutions, obtaining consent and providing clear information to users, or Refer to standards and legal frameworks on data security and privacy such as GDPR or PDPA. Therefore, I assess the legal possibility of the project as **feasible**, but it is necessary to carefully study legal issues and take preventive measures.

###### Feasible time

This is the aspect involved in determining whether the project can be completed within the desired time. Factors to consider include development time, test time, deployment time, and maintenance time. Under your constraints, the project has a tight development timeline of 12 weeks from concept to launch. This is a fairly short deadline compared to the average of mobile application development projects. This can pose risks to application quality, performance, and safety. However, there are several ways to shorten development time, such as using agile tools and techniques, prioritizing essential features eliminating unnecessary features, or hiring experts. or outside companies to support development. Therefore, I assess the project's feasible time as **unfeasible** but can be improved by adjusting the plan and work assignment.

###### Resource feasibility

This is the aspect that involves determining whether the project has the necessary resources to execute. Factors to consider include human resources, supplies, equipment, and infrastructure. According to your constraints, the project has a small development team with limited experience in mobile application development. This can cause difficulties in the expertise, capacity, and productivity of the development team. However, there are several ways to enhance the project's resource capacity, such as recruiting more experienced and high-quality staff, using tools and software that support application development, or renting cloud infrastructure and services. Therefore, I assess the project's resource capacity as **feasible**, but it is necessary to effectively manage resources and optimize resource use.

### **Create a test plan for "CampusExpense Manager".**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Case\_No** | **Purpose** | **Steps** | **Data** | **Expected result** | **Actual result** | **Status** | **Test Environment** | **Note** |
| **TC01** | Check if the new user can register with a valid username and password | 1. Open the app and select the register button.  2. Enter a unique username and valid password in the corresponding fields.  3. Click the register button. | Username: testuser  Password: Test@123 | The user will receive a confirmation message that registration was successful and be redirected to the login page |  | Pass  or  Fail | Android 10, Samsung Galaxy S10, Wi-Fi connection. |  |
| **TC02** | Check if the registered user can log in with a valid username and password. | 1. Open the application and select the login button.  2. Enter the registered username and password in the corresponding fields.  3. Press the login button. | Username: test user  Password: Test@123 | Users will receive a welcome message and be redirected to the spend management page. |  | Pass  or  Fail | Android 10, Samsung Galaxy S10, Wi-Fi connection. |  |
| **TC03** | Check if users can add new expenses with a description, date, amount, and expense type. | 1. Log into the app and select the add spending button.  2. Enter the description, date, amount, and spending type for the new spending in the corresponding fields.  3. Click the save button. | Description: Buy books  Date: October 17, 2023  Amount: 100000 VND  Expenditure type: Study | The user will receive a confirmation that the spend was successfully added and see the new spend in the spend list |  | Pass  or  Fail | Android 10, Samsung Galaxy S10, Wi-Fi connection. |  |
| **TC04** | Check if the user can edit an existing spend with a new description, date, amount, or type of spend. | 1. Log into the app and select the edit button next to the expenditure to be edited.  2. Change the description, date, amount, or spending type of the expenditure in the corresponding fields.  3. Press the button | Description: Buy discounted books  Date: No change  Amount: 80000 VND  Spending type: No change | The user will receive a confirmation that the spend has been successfully updated and will see the corrected spend in the spend list. |  | Pass  or  Fail | Android 10, Samsung Galaxy S10, Wi-Fi connection. |  |
| **TC05** | Check if users can set up monthly budgets for different types of spending. | 1. Log into the app and select the budget setup button.  2. Enter the desired amount for different spending types in the corresponding fields.  3. Tap the save button. | Budget for food: 1.000.000 VND  Budget for entertainment: 500.000 VND  Budget for studying: 800.000 VND  Budget for other expenses: 300.000 VND | Users will receive confirmation that the budget has been successfully set up and see budget amounts for spend types on the spend management page. |  | Pass  or  Fail | Android 10, Samsung Galaxy S10, Wi-Fi connection. |  |
| **TC06** | Check if users can adjust budget amounts for spending categories as needed. | 1. Log into the application and select the budget setup button.  2. Change the desired amount for one or more spending categories in the corresponding fields.  4. Click the save button. | Food budget: No change  Entertainment budget: 600.000 VND.  Study budget: No change  Budget for other types of expenses: No change | Users will receive confirmation that the budget has been successfully updated and see the new budget amounts for spend types on the spend management page. |  | Pass  or  Fail | Android 10, Samsung Galaxy S10, Wi-Fi connection. |  |
| **TC07** | Check if the app provides an overview of monthly spending, including total spending, remaining budget amount, and percentage spent by type. | 1.Log into the application and select the overview button  2. View information about monthly spending on the screen. | Total spending amount: 1.800.000 VND  Remaining budget amount: 1.200.000 VND  Percentage of spending by type:  Food and drink: 50% Entertainment: 40%  Study: 10%  Other types: 0% | Users will see information about monthly spending displayed clearly and accurately on the screen. |  | Pass  or  Fail | Android 10, Samsung Galaxy S10, Wi-Fi connection. |  |
| **TC08** | Check if users can see spending trends over time. | 1. Log into the app and select the trend view button.<br>- Select the desired period to view spending trends (e.g. week, month, year).  2. View a graph of trends spent on screen. | Period: Month  Spending trend:  January: 1.500.000 VND  February: 1.700.000 VND  March: 2.000.000 VND  April: 1.800.000 VND  May: 1.900.000 VND  June: 2.100.000 VND  July: 1.600.000 VND  August: 1.800.000 VND  September: 2.000.000 VND  October: 1.900.000 VND | Users will see a graph of monthly spending trends displayed clearly and accurately on the screen. The chart will have two axes: spending amount (VND) and time (month). The chart will have a line connecting the points representing each month's spending. The chart will have appropriate colors and legends. |  | Pass  or  Fail | Android 10, Samsung Galaxy S10, Wi-Fi connection. |  |
| **TC09** | Test if users can add recurring expenses (for example, monthly rent) with start and end dates | 1. Log into the app and select the add recurring spend button.  2. Enter the description, start date, end date, amount, and spend type for the recurring spend in the corresponding fields.  3. Press the save button. | Description: House rent  Start date: January 1, 2023  End date: December 31, 2023  Amount: 1000000 VND  Expenditure type: Housing | The user will receive a confirmation that the recurring spend was successfully added and see the recurring spend in the spend list. Recurring spending will be automatically added to the user's monthly budget for the months within the selected period. |  | Pass  or  Fail | Android 10, Samsung Galaxy S10, Wi-Fi connection. |  |
| **TC10** | Check if users can generate detailed spending reports for specific periods (e.g., monthly or yearly). | 1. Log into the application and select the Create Report button.  2. Select the desired period to create an expense report (for example, from January 1, 2023 to March 31, 2023).  3. Click the Create Report button. | Period: From January 1, 2023 to March 31, 2023  Spending during the period:  Total spending amount: 5.200.000 VND  Percentage of spending by type:  Food and drink: 40% Entertainment: 20%  Study: 30%  Housing: 10%  Other types: 0% | Users will receive detailed reports on spending for the selected period. The report will include the total amount spent, the percentage spent by type, and a pie chart showing the percentage spent by type. The report will have appropriate colors and captions. Users can save reports as PDF files or share reports via email or social networks. |  | Pass  or  Fail | Android 10, Samsung Galaxy S10, Wi-Fi connection. |  |
| **TC11** | Check if the app sends reminders or notifications when users approach or exceed budget limits for specific spending types. | 1. Log into the app and set up budgets for different spending types.  2. Add new expenses for different spending types so that the total amount spent for a certain spending type is equal to or greater than 80% of the budget for that type of spend.  3. Check to see if you receive reminders or notifications from the app | Budget for all types of expenses:  Food: 1.000.000 VND  Entertainment: 500.000 VND  Study: 800.000 VND  Housing: 1.000.000 VND.  Other other: 300.000 VND.  New spending:  Food and drink: 800.000 VND  Entertainment: 400,000 VND  Study: 700,000 VND  Housing: None  Other types: None | Users will receive reminders or notifications from the application when the total amount of a certain spending type is equal to or greater than 80% of the budget for that spending type. The reminder or notification will display the spend amount, budget amount, and percentage spent for that spend type. Reminders or notifications will have appropriate colors and sounds. Users can choose to view details or ignore reminders or notifications. |  | Pass  or  Fail | Android 10, Samsung Galaxy S10, Wi-Fi connection. |  |

Table : Test case functional Requirements

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Case\_no** | **NFR** | **Criteria** | **Scenario** | **Data** | **Expected result** | **Actual result** | **Status** | **Test Environment** | **Note** |
| **TC\_01** | Performance | The app should provide a smooth and responsive user experience, even with a large amount of expensive data | Load the app with 1000 expense records. | A database file with 1000 expense records. | The app should load within 5 seconds and display the expense data without any lag or delay. |  | Pass  or  Fail | Android 10, Samsung Galaxy S10, Wi-Fi connection | None |
| **TC\_02** | User-Friendly Interface | The user interface should be intuitive, with clear labels and easy navigation for effortless expense tracking | Navigate through the app's main features. | None | The app should have a clear and consistent layout, with labels and icons that indicate the functionality of each feature. The app should also have a navigation menu or toolbar that allows the user to switch between different screens easily. |  | Pass  or  Fail | Android 10, Samsung Galaxy S10, Wi-Fi connection | None |
| **TC\_03** | Platform Compatibility | The app should be developed for both Android and iOS platforms to reach a wider user base. | Install and run the app on different devices and operating systems. | The app's installation file for Android and iOS | The app should be compatible with various devices and operating systems, such as Android 8.0 or higher, iOS 12.0 or higher, tablets, smartphones, etc. The app should also adapt to different screen sizes and orientations without affecting the user interface or functionality. |  | Pass  or  Fail | Android 10, Samsung Galaxy S10, Wi-Fi connection | None |
| **TC\_04** | Data Security | User data, including expenses and budget information, should be securely stored and protected with encryption. Data privacy practices should be followed to ensure user information is kept confidential. | Check the security and privacy of user data in the app and the database. | A database file with some expense records and budget information. A security testing tool such as Nmap or Burp Suite. A privacy policy document for the app. | The app should encrypt the user data before storing it in the database or sending it over the network. The app should also have a password protection feature to prevent unauthorized access to the user data. The app should also follow the data privacy practices outlined in the privacy policy document, such as not sharing or selling user data to third parties, allowing users to delete their data upon request, etc. |  | Pass  or  Fail | Android 10, Samsung Galaxy S10, Wi-Fi connection | None |
| **TC\_05** | Feedback and Support | Include a feedback form within the app for users to report issues or provide suggestions.BudgetWise Solutions should actively monitor user feedback and address issues promptly. | Submit a feedback form through the app. | A feedback form with some issues or suggestions. | The app should allow the user to submit a feedback form through the app, with fields such as name, email, rating, comment, etc. The app should also send a confirmation message to the user after submitting the feedback form. BudgetWise Solutions should receive the feedback form and respond to the user within 24 hours. |  | Pass  or  Fail | Android 10, Samsung Galaxy S10, Wi-Fi connection | None |
| **TC\_06** | Monetization (Optional) | Monetization features, such as in-app ads or premium features, can be explored in future updates. Initial development should prioritize core expense management functionality. | Check the presence and functionality of monetization features in the app. | None | The app should not have any monetization features in the initial development, such as in-app ads or premium features. The app should focus on providing core expense management functionality to the user. |  | Pass  or  Fail | Android 10, Samsung Galaxy S10, Wi-Fi connection | None |
| **TC\_07** | Usability | The app should be easy to use and learn, with a user-friendly interface and clear instructions. | Perform a usability test with a sample of potential users | A usability test script with tasks and questions for the users. A usability test tool such as UserTesting or Lookback. | The app should have a high usability score, based on metrics such as task completion rate, time on task, user satisfaction, user errors, etc. The app should also receive positive feedback from the users on aspects such as navigation, layout, design, functionality, etc. |  | Pass  or  Fail | Android 10, Samsung Galaxy S10, Wi-Fi connection | None |
| **TC\_08** | Reliability | The app should be reliable and stable, with minimal bugs and errors. The app should also handle unexpected situations gracefully and recover from failures quickly. | Perform a reliability test with various scenarios and inputs. | A reliability test plan with scenarios and inputs that may cause failures or errors in the app. A reliability test tool such as Appium or Robotium. | The app should have a high-reliability score, based on metrics such as mean time between failures (MTBF), mean time to repair (MTTR), failure rate, etc. The app should also handle unexpected situations gracefully and recover from failures quickly, without losing data or functionality. |  | Pass  or  Fail | Android 10, Samsung Galaxy S10, Wi-Fi connection | None |
| **TC09** | Scalability | The app should be scalable and able to handle increasing amounts of data and users without compromising performance or functionality. | Perform a scalability test with various loads and volumes of data and users. | A scalability test plan with loads and volumes of data and users that may affect the performance or functionality of the app. A scalability test tool such as JMeter or LoadRunner. | The app should have a high scalability score, based on metrics such as response time, throughput, resource utilization, etc. The app should also maintain its performance and functionality under increasing loads and volumes of data and users, without crashing or slowing down. |  | Pass  or  Fail | Android 10, Samsung Galaxy S10, Wi-Fi connection | None |

Table :Test case non-functional Requirements

### **Why would this design work well for the "CampusExpense Manager" project?**

This design will work well for the “CampusExpense Manager” project because it meets the following requirements: Helps users manage spending effectively and save time, Has a friendly and easy-to-use interface, Has a feature to search and apply student offers for products and services, Has a feature to manage financial grants such as scholarships, tuition loans, part-time jobs, etc., Has an access feature and use school resources such as library, gym, health center, etc., Capable of operating on multiple mobile platforms such as Android and iOS.

* Helps users manage spending effectively and save time

Our design provides features such as tracking spending, setting budgets, creating spending reports, receiving spending notifications, etc., giving users a detailed overview of their situation. and their finances. Our design also allows users to enter spending quickly and accurately. These features help users control their spending and save time for other activities.

* Has a friendly and easy-to-use interface

Our design uses simple and clean interface elements, with labels and menus that are easy to understand and navigate. Our designs also use graphical elements, like charts, graphs, etc., to display spending data in an attractive and easy-to-understand way for users. These elements help users use the application smoothly and comfortably.

* Has a feature to search and apply student offers for products and services

Our design integrates with information sources about student incentives, such as discounts, freebies, etc., for products and services related to living, studying, entertainment, etc. Our design also allows users to search for offers by keyword, location, category, etc., and apply them directly from the app. These features help users save a lot of money and enjoy many benefits as students.

* Can manage financial grants such as scholarships, tuition loans, and part-time jobs...

Our design provides features such as tracking income and expenses related to financial aid grants, calculating interest rates and repayment schedules for student loans, searching and applying for scholarships, etc. These features help users manage financial benefits effectively and safely.

* Features access and use of school resources such as library, gym, health center, etc.

Our design connects to the school's student card system, allowing users to pay for school services, such as dining, laundry, printing, etc., with a student card. Our design also allows users to view student card balances and transactions, and access campus resources, such as the library, gym, health center, etc., from the app. These features help users conveniently and improve their quality of life while studying at school.

* Capable of operating on many mobile platforms such as Android and iOS

Our designs use cross-platform development tools, like Flutter, which allow you to create apps that run on both Android and iOS with a single source code. Our design also uses local databases, like SQLite, to store user data on the device and synchronize it to the cloud when an internet connection is available. These features help your application be highly compatible and work stably on many mobile devices.

* Has data synchronization and backup features

Our design uses Firebase as a cloud service to store and synchronize user data between different devices and platforms. Our design also allows users to back up their data to other cloud services, like Google Drive, Dropbox, etc. These features help users secure and protect their data from loss or damage.

* Has analysis and spending suggestions features

Our design uses Firebase Analytics to collect and analyze user spending behavior. Your design also uses machine learning algorithms to provide users with suggestions and tips on how to save money and reduce costs. These features help users improve their spending habits and achieve their financial goals.

* Has features to interact and share with the student community

Our design allows users to connect and interact with other students through features like chat, comments, reviews, and more. Our design also allows users to share expense reports, student incentives, campus resources, and more, with the student community. These features help users build relationships and support each other in financial management.

## **Activity 4: Evaluate your group's solution by comparing it with others.**

### **The SDLC model is chosen over other SDLC models.**

The Rapid Application Development model was chosen over other SDLC models because it is suitable for the nature and scale of the "CampusExpense Manager" project. This project is a mobile application for students, with high demands for novelty, creativity, and user-friendliness. This project may also encounter many changing requirements from customers and users during development. Therefore, the Rapid Application Development model is the best choice because it allows the development team to communicate and collaborate closely with customers and users, divide the project into concise and flexible phases, test and improve Continuous products, and receive early feedback. My team chose the Agile model, which has the advantage of being able to adapt quickly to the changes in the requirements of the customers, enhance the collaboration between the team members and the customers, and improve the quality of the product by testing continuously and getting early feedback. However, the Rapid Application Development model also has drawbacks such as difficulty in managing budget and time, requiring customers to participate actively in the development process, and having the risk of losing focus on the important features of the product.

Other SDLC models, such as Waterfall, V, Spiral, Circle model, and Growth models may not be suitable for the "CampusExpense Manager" project for the following reasons:

* + The Waterfall model is a sequential model, in which the stages of the project are performed in a fixed order and there is no interaction between the stages. This model may not be able to adapt to the changes in the requirements of the customers and users, causing delays and waste of resources. This model may also not ensure the high quality of the product because testing is only done at the final stage of the project.
  + The V model is a variant of the Waterfall model, in which the testing stages are linked to the corresponding development stages. This model may improve the quality of the product by testing the requirements, design, and code at each stage. However, this model also has similar drawbacks to the Waterfall model, such as being inflexible to changes in requirements, having no communication between customers and the development team, and having no intermediate products to test.
  + The Spiral model is a combination of Circular and Waterfall models, in which the stages of the project are repeated several times with increasing levels of detail. This model may help the development team deal with risk and change requirements effectively, produce high-quality products, and meet customer needs. However, this model also has drawbacks such as requiring a lot of time and resources to perform risk analysis and evaluation activities, being difficult to estimate the budget and time for the project, and being unsuitable for small or clear-requirement projects.
  + The Circular Model is an iterative model in which project phases are repeated multiple times until the desired results are achieved. This model can cause ambiguity and lack of clarity regarding project goals and progress. This model can also cause duplication of work and inefficiencies in resource management.
  + The Growth Model is a hybrid model between the Falling Water model and the Circle model, in which the project stages are carried out in a fixed sequence but can be repeated if necessary. This model can give development teams more flexibility in adapting to changing customer and user requirements, but can also cause complexity and uncontrollability in project management.

### **Selected tools compared to other development tools.**

1. GitHub compared to other source code management tools

GitHub is an online source code hosting service based on the Git version control system. GitHub has advantages like allowing development teams to collaborate and synchronize their source code easily and securely, supporting features like branches, merges, pull requests, etc., providing tools like bug tracking boards, wikis, static websites, etc., integrating with many other services and tools, like Firebase, Visual Studio, Slack, etc. However, GitHub also has disadvantages such as requiring an internet connection to use, may encounter problems with source code conflicts when working with many people, and has limited storage capacity and privacy. for free repositories.

1. Microsoft Project compared to other project management tools

Microsoft Project is a popular and professional project management software. Microsoft Project has advantages such as allowing the development team to plan, monitor, and control project activities effectively and in detail, supporting features such as Gantt charts, network charts, and charts. schedule, etc., providing tools like reporting, risk analysis, budget forecasting, etc., and integrating with other Microsoft products, like Office 365, SharePoint, Teams, etc. However, Microsoft Project also has disadvantages such as high cost and requiring a license to use, having a complex interface that is difficult to use for beginners, and not being well-compatible with operating systems. and other tools.

1. Android Studio compared to other IDEs for Android development

Android Studio is an official and free IDE for Android application development. Android Studio has advantages such as providing a development environment fully integrated with tools and resources for Android and supporting features such as intelligent source code editing, application debugging and testing, and design. the drag-and-drop user interface, etc., provide tools like Android virtual machine (AVD), Firebase Assistant, APK Analyzer, etc., and integrates with popular frameworks and libraries for Android, like Jetpack Compose, Kotlin Coroutines, Dagger Hilt, etc. However, Android Studio also has disadvantages such as requiring a high-configuration computer to run smoothly, encountering errors and problems when updating versions, and only supporting application development for the Android platform.

1. Canva compared to other graphic design tools

Canva is a free and easy-to-use online graphic design service. Canva has advantages such as allowing users to create high-quality graphic products, such as logos, charts, posters, etc., supporting features such as drag and drop, editing, adding text, etc., providing free or low-cost templates, images, icons, fonts, etc., for users to choose from, integrating with other services and tools, like Google Drive, Dropbox, Facebook, Instagram, etc. However, Canva also has disadvantages such as requiring an internet connection to use, limited features and resources for free users, and may not be able to meet the needs of professional and complex graphic design. junk.

1. draw.io compared to other diagramming tools

draw.io is a free and open-source online diagramming service. draw.io has advantages such as allowing users to create different types of diagrams, such as data flow diagrams, class diagrams, state diagrams, etc., supporting features such as drag and drop, editing, adding text, etc., providing templates and illustrations for different types of diagrams, store and back up data on various cloud services, like Google Drive, Dropbox, OneDrive, etc. However, draw.io also has disadvantages such as requiring an internet connection to use (unless downloading the desktop version) and having an interface that is not too user-friendly and attractive.

# **Conclusion**

In this report, we have introduced the "CampusExpense Manager" mobile application, a tool that helps students manage their expenses effectively and save time. We have presented the details of the activities and results of the development process, including the use of appropriate software development tools and techniques to deploy the solution, the analysis of business problems to come up with proposed solutions, the design of the user interface and business logic of the application, and the testing of the quality and performance of the application.

The "CampusExpense Manager" application has some novel and valuable features for users and the community, such as:

* Providing a friendly and easy-to-use interface, helping users manage their expenses quickly and conveniently.
* Operating on multiple mobile platforms such as Android and iOS, allowing users to use the application on any device they have.
* Offering a feature to search and apply student discounts for various products and services that users use, helping users save costs and increase the value of their money.
* Offering a feature to manage financial aid such as scholarships, loans, part-time jobs, etc., helping users monitor their financial situation and plan to pay off debts or save reasonably.
* Offering a feature to access and use campus resources such as libraries, gyms, health centers, etc., helping users enjoy the facilities and services of the school without spending time or money.

To further develop this application, we propose the following directions:

* Adding new features such as integrating with electronic payment applications, allowing users to share budgets and expenses with friends or family, and providing suggestions and advice on how to save money and manage finances effectively.
* Improving the user interface by adding attractive elements such as icons, colors, sounds, animations, etc.
* Enhancing security and data protection by using encryption, authentication, and compliance methods with information security regulations.
* Conducting surveys and user testing to receive feedback and evaluation on the application, as well as to improve and enhance product quality.

We hope that the "CampusExpense Manager" application will bring many benefits to users, especially students, in managing their expenses and finances. We also look forward to receiving support and feedback from readers to improve the application further. Thank you.

# **References**

Anon, 2022. *What is a Software Process Model?.* [Online]   
Available at: https://www.visual-paradigm.com/guide/software-development-process/what-is-a-software-

Anon, 2023. *www.builtin.com.* [Online]   
Available at: https://builtin.com/authors/artem-oppermann

Naidu, N., 2018. *Software Engineering | Agile Software Development.* [Online]   
Available at: https://www.geeksforgeeks.org/software-engineering-agile-software-development/

Pal, S. K., 2018. *Software Engineering | Spiral Model.* [Online]   
Available at: https://www.geeksforgeeks.org/software-engineering-spiral-model/

Rajkumar, 2019. *Software Testing Material.* [Online]   
Available at: https://www.softwaretestingmaterial.com/sdlc-software-development-life-cycle/