

Software Requirements Specification

for

CalTacToe

Version 1.0

Prepared by G01_01

Client: Mohd Faiz Mohd Subri

14 March 2024

Table of Contents

Table of Contents	ii
Revision History	iii
1. Introduction.....	4
1.1 Purpose	4
1.2 Document Conventions	4
1.3 Intended Audience and Reading Suggestions.....	4
1.4 Project Scope	5
1.5 References	5
2. Overall Description	5
2.1 Product Perspective	5
2.2 Product Features	6
2.3 User Classes and Characteristics	7
2.4 Operating Environment	7
2.5 Design and Implementation Constraints.....	7
2.6 User Documentation	8
2.7 Assumptions and Dependencies	8
3. System Features	8
3.1 System Feature 1	8
3.2 System Feature 2 (and so on)	19
4. External Interface Requirements	22
4.1 User Interfaces.....	22
4.2 Hardware Interfaces.....	26
4.3 Software Interfaces	26
4.4 Communications Interfaces	26
5. Other Nonfunctional Requirements.....	26
5.1 Performance Requirements.....	26
5.2 Safety Requirements.....	26
5.3 Security Requirements.....	27
5.4 Software Quality Attributes.....	27
6. Other Requirements	28

Appendix A: Glossary.....	28
Appendix B: Analysis Models	28
Appendix C: Issues List.....	29

Revision History

Name	Date	Reason for Changes	Version
1. CalTacToe V.1	14 March 2024	Initial release	1.0
	16 March 2024	Formatting Correction	1.1

1. Introduction

1.1 Purpose

The Software Requirements Specification or SRS is needed in any software development project because it serves several crucial purposes as:

- It **establishes agreement** between customers and developers regarding what the software product is expected to do.
- **Risk reduction** by providing rigorous assessment of requirements before the start of any sprint to help reduce the need for later redesign.
- SRS provides a realistic basis for **estimating product costs, risks, and schedules**.
- It acts as a **written agreement** between you and your development team members, **ensuring alignment on project goals**.
- The SRS also **eliminates the miscommunications and misunderstandings** among team members by clearly defining technical details about the software.

1.2 Document Conventions

N/A

1.3 Intended Audience and Reading Suggestions

AUDIENCE	READING SUGGESTION
developers (team members)	To use this documentation to understand and avoid miscommunication about the technical details about the software.
project manager	To approve overall contents in this documentation.
users	To know about the product that they are going to get and its information.

testers	Use this documentation for user manual and guidance.
documentation writers	The one who manages and writes this documentation.

1.4 Project Scope

As far as we know, the project management scope means the features and functions of a product or the extent of work required to complete a project. The scope keeps us stay in the roadmap, create effective Stakeholder Management, budgeting and scheduling and preventing scope creep. So, for our project, the detailed project scope statement is as written below:

- Timeline: 11 weeks (4 weeks for Sprint 1 and 2, and 3 weeks for final Sprint)
- Budget: none
- Assigned Tasks: assigned fairly for each of our member consist of leader, programmer and documenter.
- Project Stakeholder(s): Mohd Faiz bin Mohd Subri (Product Owner)
- Workflow Strategies: using SCRUM practices

1.5 References

We referred to one and only website that explaining detailly about SRS and show some examples how to write it. The website is:

<https://jelvix.com/blog/software-requirements-specification>

2. Overall Description

2.1 Product Perspective

A brand-new product or mobile app specifically, where our own made app tick all the boxes; requirement we need that already-existing similar app like calculator app but better:

- Create more optimized app on performance aspect for faster loading time and smoother interactions to ensure customers' satisfactory and encourage their revisability.
- Taking advantage over mobile apps that are omnipresence because of their accessibility at everywhere and anytime unlike desktop apps.
- Making app for our customer to provide dedicated platform for their own brand; CalTacToe!
- A self-made app enables us to create custom interactions based on our customer's requirements/goals like the customer service button to straightaway contact the developer.
- allows us to tailor features specifically and specially to our customers' needs.

2.2 Product Features

We have list down all of our apps' features that meet our customer requirements during our Google Meet interview. The apps/products' features that are now in our product backlog are as below:

1. As a user, I also want to perform complex arithmetic operations like percentage besides the basic one (addition, subtraction, multiplication, and division) using the calculator.
2. As a user, I want to see the result of my calculations displayed clearly on the screen.
3. As a user, I want to be able to clear the current calculation and start a new one.
4. As a user, I want to be able to exit out of the calculator with off button.
5. As a player, I want to play tic-tac-toe against another player.
6. As a player, I want the game board to be displayed clearly, showing the current state of the game.
7. As a player, I want to be notified when the game ends (win, lose, or draw).
8. As a user, I want buttons or controls that are easy to understand and interact with.
9. As a user, I want to provide feedback or report issues related to the app.
10. As a user, I want my app to have a standout icon.

2.3 User Classes and Characteristics

This apps have two types of end-user as shown in table below:

USER CLASS	CHARACTERISTICS
App Owner	The one who own the app.
Second player	The one who compete first player in Tic Tac Toe mode.

2.4 Operating Environment

Obviously on mobile platform. The app is supported by Android, the most used operating system and also used in every apps nowadays, but not iOS.

2.5 Design and Implementation Constraints

I am going with minimalistic design approach. For example, use less buttons, pages and only use one color theme. Minimalistic design has become increasingly popular in mobile app development for several compelling reasons:

- Create a balance mobile app between functionality and aesthetic, resulting in effective, and eye pleasing.
- Easy navigation, meaning we respect our customers' time, attention, and preferences.
- Removing unnecessary elements and simplifying them like image buttons, graphics and such means faster loading time, lower resources consumption, enhance performances, and most importantly, improve efficiency.
- Lastly, the simplicity will encourage the users to be more engage, and only use this calculator instead of buying the physical one that is more expensive.

For the implementation of design constraint, we think there are not much to be expected as the minimalistic designs are most preferred nowadays. We can only think of two constraints that might come to our mobile app like:

- Difference people means difference taste. Some people don't appreciate minimal design but more into mobile apps with cluttering buttons, pages, details and more.
- Minimalistic design has less accessibility like the icon without text will confuse some users, the not-much-to-do app will irritate users who hope for more functionality, and others.

2.6 User Documentation

SOFTWARE DOCUMENTATION		DESCRIPTION
SDP		Documenting software development plan.
SRS		Documenting software requirement specification.
SDD		Documenting software design.
STD (Software Test Documentation)	STP	Documenting software test plan.
	STD	Documenting software test description.
	STR	Documenting test result.
SUD		Documenting software user manual.

2.7 Assumptions and Dependencies

N/A

3. System Features

Organizing the functional requirements for the product by system features, the major services provided by the product.

3.1 User Features

3.1.1 As a user, I also want to perform complex arithmetic operations like percentage besides the basic one (addition, subtraction, multiplication, and division) using the calculator.

System Name	CALTACTOE
Team Id	G01_01
Number #	1
Title	ADD ARITHMETIC OPERATIONS (CALCULATOR)
Description As I want to So that	A user perform arithmetic operations I can get the result I want
Priority	High
Iteration / Sprint	1
Status (Planning / In Progress / Complete)	In Progress
Acceptance Criteria / Confirmation	
<ul style="list-style-type: none">• Enter operands• Enter operators	
Task	
<ol style="list-style-type: none">1. Analyze feature details2. Design UI3. Coding	

3.1.2 As a user, I want to see the result of my calculations displayed clearly on the screen.

System Name	CALTACTOE
Team Id	G01_01
Number #	2
Title	SHOW CLEAR RESULT (CALCULATOR)
Description As I want to So that	A user Clearly see result on calculator screen I can know the result better
Priority	High
Iteration / Sprint	1
Status (Planning / In Progress / Complete)	In Progress
Acceptance Criteria / Confirmation	
1. Tap on equal button 2. The result should clearly displayed on black calculator screen	
Task	
1. Analyze feature details 2. Design UI 3. Coding	

3.1.3 As a user, I want to be able to clear the current calculation and start a new one.

System Name	CALTACTOE
Team Id	G01_01
Number #	3
Title	ADD CLEAR BUTTON (CALCULATOR)
Description As I want to So that	A user Clear calculator screen I can perform another arithmetic operations
Priority	High
Iteration / Sprint	1
Status (Planning / In Progress / Complete)	In Progress
Acceptance Criteria / Confirmation	
1. Tap “clear” button 2. The result screen should all-clear	
Task	
1. Analyze feature details 2. Design UI 3. Coding	

3.1.4 As a user, I want to be able to exit out of the calculator with off button.

System Name	CALTACTOE
Team Id	G01_01
Number #	4
Title	ADD OFF BUTTON (CALCULATOR)
Description As I want to So that	A user Tap off button I can exit from the app
Priority	High
Iteration / Sprint	1
Status (Planning / In Progress / Complete)	Planning
Acceptance Criteria / Confirmation	
1. Hit off button 2. User brought out from the app to do another things / open another app	
Task	
1. Analyze feature details 2. Design UI 3. Coding	

3.1.5 As a player, I want to play tic-tac-toe against another player.

System Name	CALTACTOE
Team Id	G01_01
Number #	5
Title	ADD SECOND PLAYER (TIC TAC TOE)
Description As I want to So that	A first player Add second player I can compete with my family / friend member
Priority	High
Iteration / Sprint	2
Status (Planning / In Progress / Complete)	Planning
Acceptance Criteria / Confirmation	
<ul style="list-style-type: none">• Player 1 (X's) can enter name• Player 2 (O's) can enter name	
Task	
<ol style="list-style-type: none">4. Analyze feature details5. Design UI6. Coding	

3.1.6 As a player, I want the game board to be displayed clearly, showing the current state of the game.

System Name	CALTACTOE
Team Id	G01_01
Number #	6
Title	DISPLAY GAME BOARD (TIC TAC TOE)
Description As I want to So that	A player clearly see board on screen We (two players) can clearly see what going on / where to mark
Priority	High
Iteration / Sprint	2
Status (Planning / In Progress / Complete)	Planning
Acceptance Criteria / Confirmation	
1. Board clearly displays on screen 2. Player 1 and 2 able to mark with X and O respectively each turn	
Task	
1. Analyze feature details 2. Design UI 3. Coding	

3.1.7 As a player, I want to be notified when the game ends (win, lose, or draw).

System Name	CALTACTOE
Team Id	G01_01
Number #	7
Title	NOTIFY GAME RESULT (TIC TAC TOE)
Description As I want to So that	A player see the end-game result we can know the winner or the game is draw.
Priority	High
Iteration / Sprint	2
Status (Planning / In Progress / Complete)	Planning
Acceptance Criteria / Confirmation	
1. When the game met criteria for a winner and if no grid left to mark anymore, 2. It will pop out the result board at the center of screen.	
Task	
1. Analyze feature details 2. Design UI 3. Coding	

3.1.8 As a user, I want buttons or controls that are easy to understand and interact with.

System Name	CALTACTOE
Team Id	G01_01
Number #	8
Title	OPTIMIZE BUTTONS AND CONTROLS (ALL)
Description As I want to So that	A user see my app's buttons are attractive and aesthetic It can sooth my eyes and heart.
Priority	Low
Iteration / Sprint	3
Status (Planning / In Progress / Complete)	Planning
Acceptance Criteria / Confirmation	
1. Buttons and layout perfectly organized and sized 2. Apply minimalistic color and theme	
Task	
1. Analyze login details 2. Design UI 3. Coding	

3.1.9 As a user, I want to provide feedback or report issues related to the app.

System Name	CALTACTOE
Team Id	G01_01
Number #	9
Title	ADD CUSTOMER SUPPORT BUTTON (ALL)
Description As I want to So that	A user Be able to contact developer team I can report issues / give any feedback about the app
Priority	Low
Iteration / Sprint	3
Status (Planning / In Progress / Complete)	Planning
Acceptance Criteria / Confirmation	
1. Tap 'customer service' button 2. Able to send email to developer team via gmail	
Task	
1. Analyze feature details 2. Design UI 3. Coding	

3.1.10 As a user, I want my app to have a standout icon.

System Name	CALTACTOE
Team Id	G01_01
Number #	10
Title	ADD ICON (ALL)
Description As I want to So that	A user See that my app have a standout icon It can reflect its main use / purpose
Priority	Low
Iteration / Sprint	3
Status (Planning / In Progress / Complete)	Planning
Acceptance Criteria / Confirmation	
<ol style="list-style-type: none">1. Tap 'order history' button2. Browse the history of the products you used to buy	
Task	
<ol style="list-style-type: none">1. Analyze feature details2. Design UI3. Coding	

4. External Interface Requirements

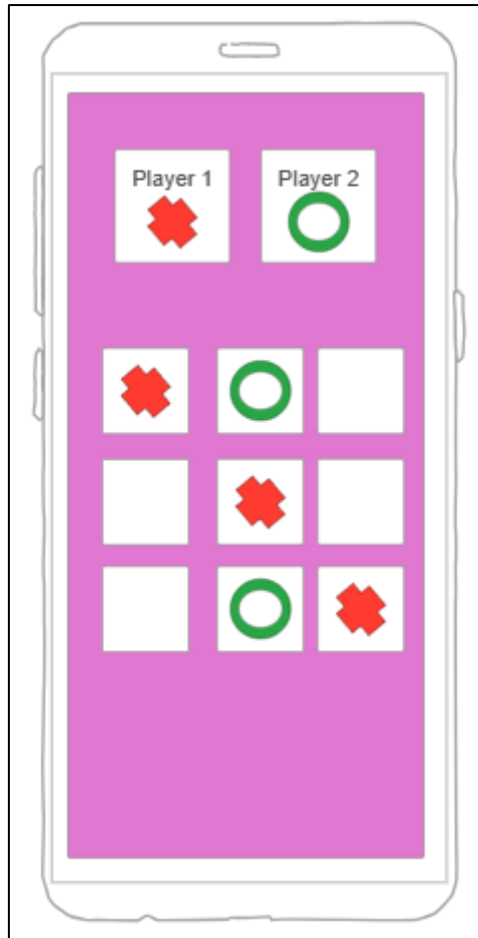
4.1 User Interfaces



- 1) **calFragment**: a main page where user can use calculator to perform arithmetic functions like addition, subtraction, division and multiplication and it supports decimal points.



2) **playerFragment:** a page a player 1 and player 2 have to sign up name before being brought to the tic tac toe 'arena'.



3) **boardFragment**: a tic tac toe page where players compete on game board to see which player matched three box first.



4) **resultFragment:** a page (dialog box to be precise) the system announce the result of the game.

4.2 Hardware Interfaces

N/A

4.3 Software Interfaces

- Android: operating system that manages resources and provide essential services for mobile app.
- Android Studio: where the mobile app is designed and being coded.

4.4 Communications Interfaces

- Email: the store owner's email linked to this mobile app that act as customer support to receive questions or feedbacks from their customers.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

With minimalistic approach in every aspect (fewer complex buttons, simple codes, less pages, etcetera), this app is being expected to run smoothly and response quickly at the command of users. The explanation is on 2.6.

5.2 Safety Requirements

The safety requirements here mean what are to take to make the data in mobile app doesn't have the risk to loss or damage. So, we hope to implement the use of database in the future, or in Android Studio case, SQLite. SQLite is an open-source relational database that is used to perform database operations on android devices such as storing, manipulating, or retrieving persistent data from the database. It is also powerful yet low-level because it requires manual handling. So, to add the efficiency and the convenient way to work with this database, we will consider using Room.

5.3 Security Requirements (Not implemented yet)

Security requirements here mean what are to take to make this mobile app has specific user authentication. So, we make sure that the admin or the app owner has their own constant or

premade password and username to login to ensure only them can have the ability to CRUD (create, read, update and delete) products from the store's database/list. The keyword is powerfully strong because it is a combination of several uppercase and lowercase letters with some symbols and numerical values. A password with this length and complexity will take someone 10 years to crack it if they do the try and error tricks non-stop. If we ever put the requirement for twelve characters, it will be 200 years!

5.4 Software Quality Attributes

1. **Usability:** We will make sure our users can easily interact with the app and achieve their goals. It is to increase overall user experience, ease of navigation, and satisfaction.
2. **Familiarity:** Ensure that users quickly understand our app's user interface (UI). Using universally familiar patterns and consistent design elements will enhance usability.
3. **Efficiency:** Our users should achieve their objectives efficiently and having minimize unnecessary steps and streamline workflows.
4. **Error-Free Interaction:** Our app should work as expected, without unexpected errors or confusing behaviour.
5. **Reliability:** Our app to consistently perform its intended functions without unexpected failures.
6. **Accessibility:** The app should be available when our users need it.
7. **Responsiveness:** It should respond promptly to our user actions.
8. **Robustness:** Because of the low resource consumption and less complex functions, we expected that our app should handle errors gracefully and recover without crashing.
9. **Updatable and Maintainable:** we will make sure to do regular updates or prompt updates if received any feedback to ensure reliability.
10. **Scalability:** We design our app to handle increased load without compromising reliability.

All of the above achieved when we implemented minimalistic approach as explained in 2.6.

6. Other Requirements

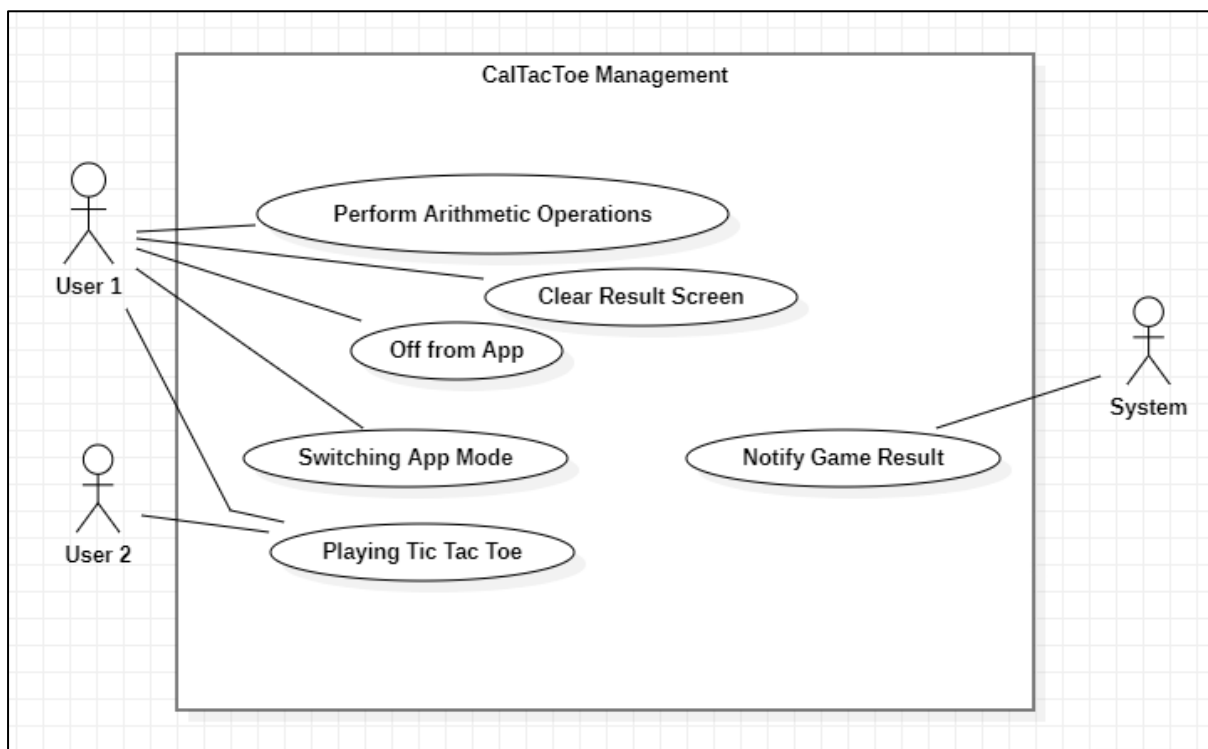
N/A

Appendix A: Glossary

A list of short-form or jargon words that are most used and playing important parts in this documentation.

NO	TERM	DEFINITION
1	SRS	Abbreviation for Software Requirements Specification.
2	app owner	Admin of the app.

Appendix B: Analysis Models



Appendix C: Issues List

The issues we are concerned when working on this mobile software development are:

- Mobile programming is new to us and tinkering with Android Studio is quite challenging. Hope we can manage and being more skilled over time.
- The more codes added the more complex it is to maintain the app.