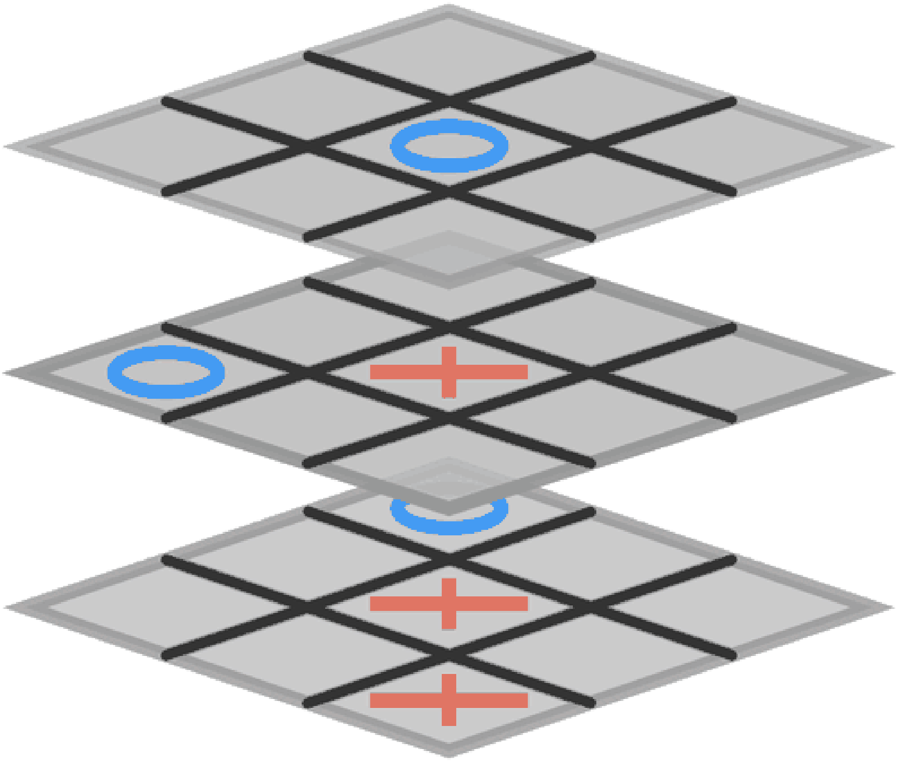


3D-Tic-Tac-Toe

Group 30
Prepared by
Hasan Ali, Umer Qazi, Syed Raza and Claudio Jimenez
***for use in* CS 440**
at the
University of Illinois Chicago

Fall 2020

Table of Contents

List of Figures

List of Tables

Environment of the current System

Project Description

Project Overview

3D Tic-Tac-Toe is an extension of the century old game of Tic-Tac-Toe. This game will be available on both mobile software (iOS and Android) and as a website for PC users. Each player will create an account with a unique user ID. This will track their progress in the game and their achievements. The new game will have all the rules from the original game and some more. 3D Tic-Tac-Toe will consist of up-to 4 players with every player trying to win the game before others. The game surface will be in a cube shape resulting in 6 total surfaces where the game can be played. A player needs to win on 2 total surfaces to be considered a winner. The game will have powerups/charms to make the game more exciting. It will be a free-to-play game with players earning or purchasing different power ups and avatars for their gameplay.

The Purpose of the Project

The User Business or Background of the Project Effort

There are experienced fans of the original Tic-Tac-Toe game worldwide. Many adapt to the simple rules of the game and have an interest in gaming at a competitive level to where the stakes are higher and more than just 2 players can be involved. The online gaming business is a very large market that modern gamers all have become a part of. Allowing gamers to play with each other via multiplayer online gaming opens up the potential of the game to reach great heights. There is always a demand for new games and 'fan favorite' games that have a modern twist to it. We intend to sell this directly to gamers as the gaming industry is a large and growing one.

Goals of the Project

Our 2 primary goals are challenge fans of the original Tic-Tac-Toe game and allow multiple players to be involved. We want to bring a fresh new look to all the fans of the original game with new features, powerups, and perks as you continue your journey on this game. This will create an enjoyable multiplayer atmosphere that is popular in today's modern games as well as including perks that allow your player to improve and be rewarded. Allowing players to have these new features will allow the players to play the refreshed 3D version for a much longer time span than the original 2d tic-tac-toe. Another goal of ours is to increase game platform subscribers which will ultimately increase game revenues. This is vital to having 3D tic-tac-toe being a success as our game design should retain the platform subscribers and increase over time.

Measurement

We will measure the use of our product via feedback and reviews given by our customers. As we plan to have our game implemented in the apple store, google play store, and on our website, users will be able to provide real time feedback on the game to express how they are feeling. These ratings out of 5 stars will be a key indicator if our users are truly enjoying the game and all of its features. This data will allow us to determine what features of the game the players like and don't like, as well as informing us on any missing content they would like to have included in the game. After completing a game, the users will see a small pop up on the side that will ask if the wish to provide feedback/review on the game. We will know when these metrics will be achieved when reviews increase, and player subscribers increase by 10%.

The Scope of the Work

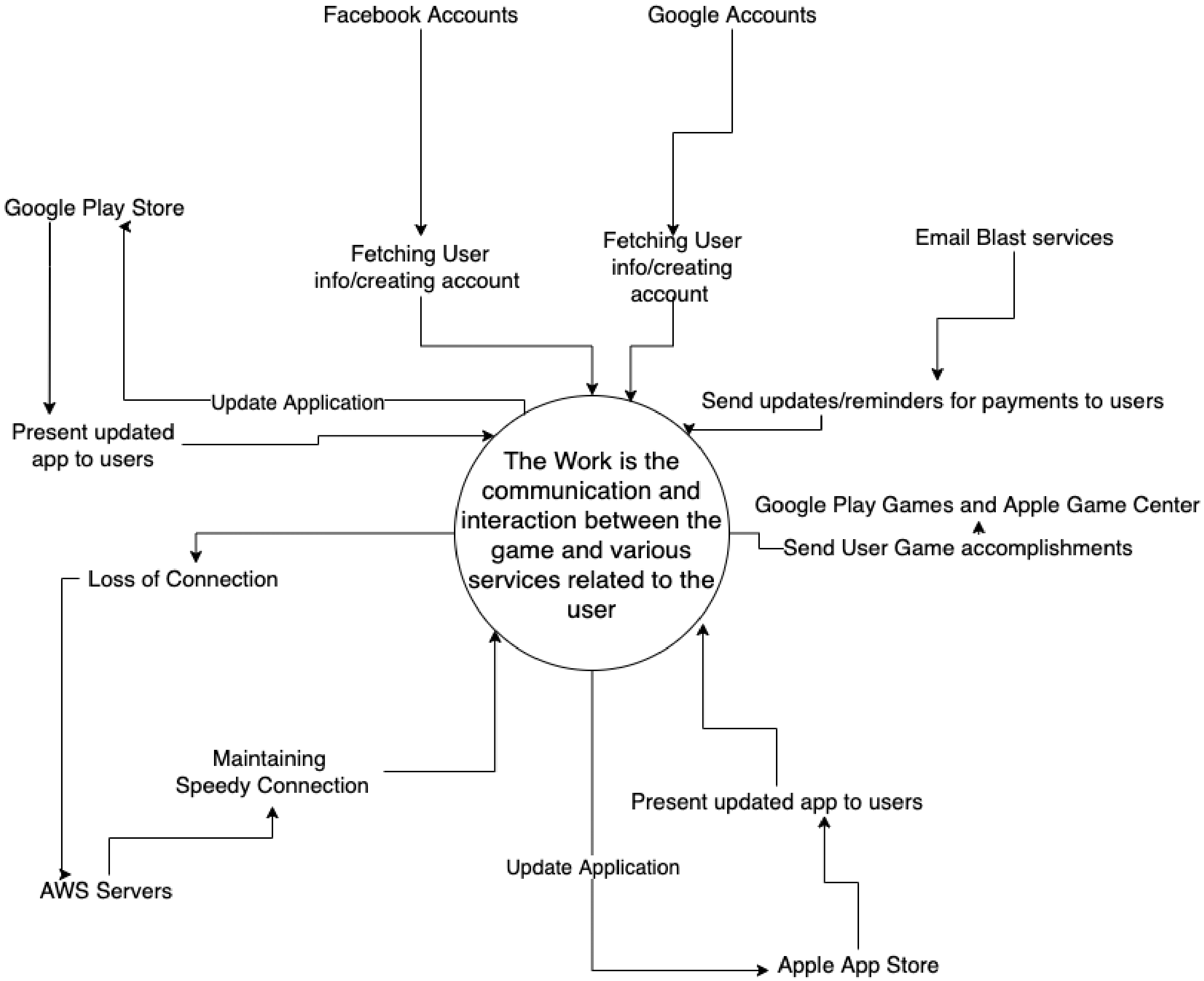
The “work” is the communication and interaction between the game and various services related to the user.

The Current Situation

The client, Board Games 2.0 INC, would like to explore this specific field as they see it having potential. This would build upon their series of n-D games such as 4-DChess and 4-DCheckers with 3-D Tic Tac Toe.

As for the current situation for the game itself, it is simply the basic game that can be played anywhere with pencil and paper. Its origination can be dated back to 1300 BC and yet it has stood against the test of time with children today playing the same game. While there are a multitude of individuals that have created this game, it cannot be attributed to any one single entity.

The Context of the Work


Facebook Accounts
Google Accounts
Google Play Store
Fetching User info/creating account
Fetching User info/creating account
Email Blast services
Update Application
Send updates/reminders for payments to users
Present updated app to users
The Work is the communication and interaction between the game and various services related to the user
Google Play Games and Apple Game Center
Send User Game accomplishments
Loss of Connection
Maintaining Speedy Connection
Present updated app to users
Update Application
AWS Servers
Apple App Store

The context of the work diagram

Work Partitioning

Business Event List

Event Name	*Input and Output*	*Summary*
1. Update application for Android	*Publish updated app to Google Developer Console (out)* *Allow users to update app on their phone(in)*	*Update the code with new features, to account for policy change etc. Need to tell Google to update for all*
2. Update application for Apple	*Publish updated app to Apple Developer (out)* *Allow users to update app on their phone(in)*	*Update the code with new features, to account for policy change etc. Need to tell Apple to update for all users*
3. Facebook integration with user's Accounts	*User profile info (in)*	*Whenever a user signs in with a Facebook account, will have to re-direct to their website*
4. Email Service blast periodically	*Send reminder emails for payments/new features (in)*	*Keep users updated with new features for the game and entice them back in*
5. Problem with AWS servers	*Report problem of servers (out)* *Fix problem/find out who's problem it is*	*Servers needs to stay up 24/7 so problem needs to be fixed quickly*
6. Google integration with user's Accounts	*User profile info (in))*	*Whenever a user signs in with a Google account, will have to re-direct to their website.*
7. Update game info to Facebook account for user to share	*Send game accomplishments to Facebook (out)*	*Give the user the ability to share their accomplishments through Facebook*
8. Google Play Games and Apple Game Center update to new rankings	*Send game accomplishments to Google/Apple (out)*	*Using Google/Apple services, users can easily compare their accomplishments with their friends*

Competing Products

There are different attempts at a game of 3-D Tic Tac Toe but none as elaborate as this one. One simple search comes with multiple examples of these attempts.

However, they are played on various websites with multiple other games and no features for collaboration/competition. It is not a complete experience for the user, rather they are all very basic in nature with subpar graphics. Through various communication features, customization features, and Artificial Intelligence we will be revamping the whole game with a new set of rules while retaining the simplistic foundational nature which made it so enjoyable in the first place. It will give users the experience of a full-fledged game while also challenging themselves to get better at it.

The Scope of the Product

3D Tic-Tac-Toe is a modified version of the traditional 2D game, the proposed product addressed the need of the online game business to be constantly innovating and developing new products, of providing entertainment, bringing the player's community together and encouraging interaction and competition among gamers. 3D Tic-tac-toe would achieve this work by creating a game platform (online, android and iOS Apps) with a front-end interface where user interact with the system and a back-end server that connects the players, runs the games logic, provides AI players, keeps gamers scores, their account info, accomplishments, provides messenger services and system/game help. System maintenance and updates to the game will performed on a schedule by the system managers.

Scenario Diagram

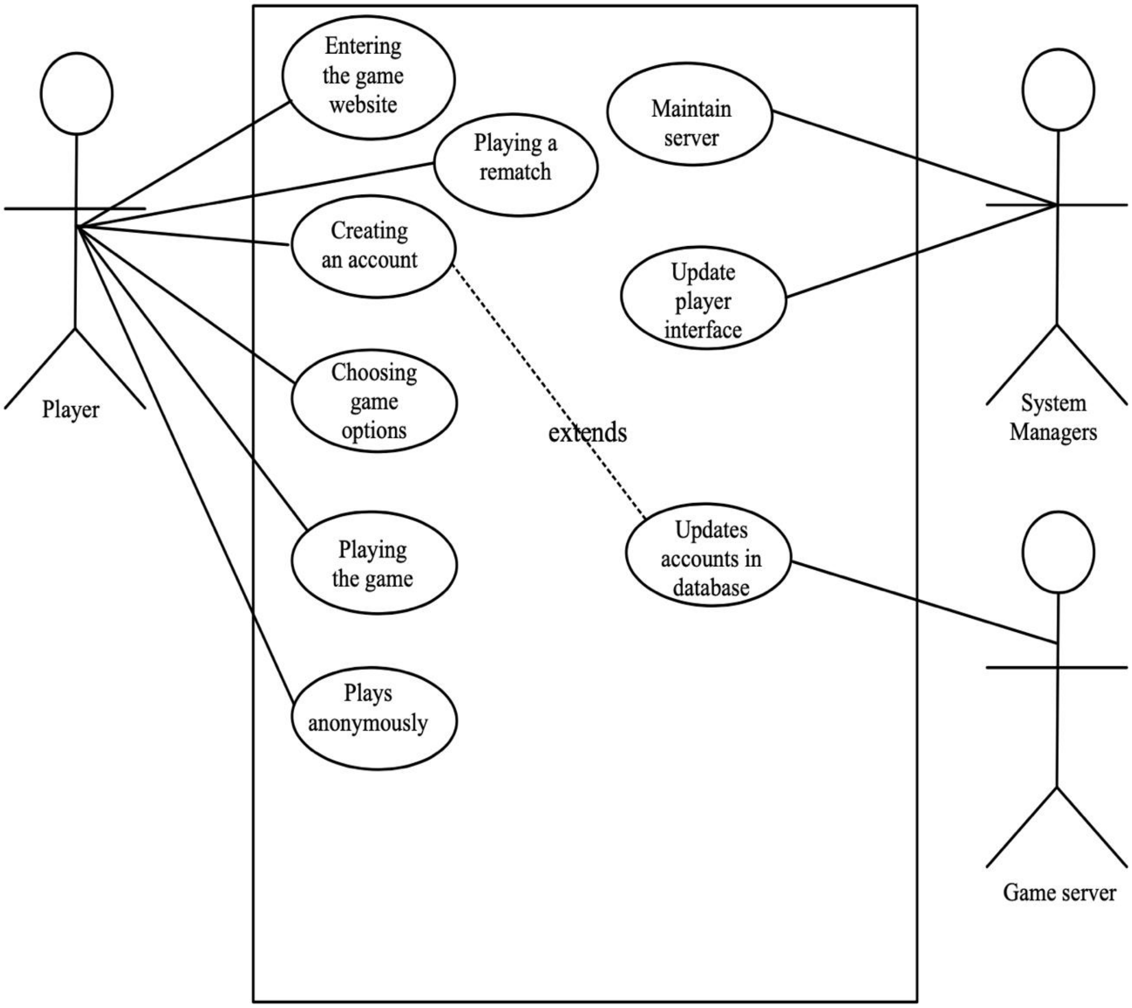


Figure 4.1 Scenario Diagram

Product Scenario List

Scenarios	Users
Pre-Game Scenarios	
1. Entering the game website	Players
2. Creating an account	Players
3. Playing anonymously (without account)	Players
4. Choosing game options	Players
Gameplay Scenarios	
5. Playing the game	Players
6. Playing a rematch	Players
Post-game Scenarios	
7. View profile and accomplishments 8. Have ability to share on social media	Players Players, System managers
System Maintenance Scenarios	
9. Maintain server	System managers
10. Update player interface	System managers

Individual Product Scenarios

1. **Entering the game website:** If Joe want to play online, he will open a web browser and enter the game website address via email invitation or advertised link. The system will serve a welcoming page to Joe with the options to create an account, login or play anonymously. Joe will know that he is in the right page by seeing the company logo and game name.

2. **Creating an account:** On the welcoming screen Mary will click on create account button, she will be redirected to another screen where she would be asked to enter her name, chosen username, phone number for account recovery(optional), password, credit card information and billing address(optional) Once Mary enters her information, she will have the option of creating a game avatar. She would have the choice of customizing the avatar's face, body and attire. The game server will add the account to the database and serve Mary with the game lobby screen.

3. **Playing anonymously:** On the welcoming screen Tom will click on the play anonymously button. The server will redirected Tom to the game lobby screen. The game server will automatically generate a username for Tom that he can only use

once. Tom would not be able to take advantage of any of achievements or point system of the game. Tom is ready to set his desired game options and start playing.

4. **Choosing game options:** Hana would be presented with the game lobby screen, she would have to set the following options,

    i. Type of game: Hana could play against other players, or against AI. She could choose to play a ranked tournament, or a practice run.

    ii. She could play with up to 4 players.

    iii. Hana must choose a difficulty level between easy, medium, hard or expert. Only on ranked games Hana will earn points. The higher the difficulty the more point the Hana earns. The system would place Hana with other players with similar points and ranking.

    Once Hana is finished setting these options, she would be able to start the game by pressing the start game button. The system would start looking for other players depending on her chosen settings and would display to Hana an approximate wait time. Hana would decide if she wants to wait the displayed time or change her options. Once the system finds players with similar settings as Hana, they will be bundle into game group and sent to the gameplay screen.

5. **Playing the game:** Harman is now in the gameplay screen. Harman will see a 3D game board were players can enter their move via the player interface, his score, ranking and power ups. Harman will play according to the game rules. Once Harman wins, he would get his awarded points, badges and avatar options (he would only have these awards if he has a game account) If the system declares a game tie, there would be no awarded points or badges. In both cases Harman would be redirected to the game lobby to start another game or quit.

6. **Playing a rematch:** After the system declares a winner or a tie game Gloria would be redirected to the game lobby screen where she and the other players are bundle as a game group, here Gloria can choose to quit the game group or choose a rematch. If Gloria decides to quit the game but the other players decide to stay the new game would only have the remainder players without Gloria. If she decides to stay in the group and play a rematch, her game group would be redirected to the gameplay screen for another match.

7. **View profile and accomplishments:** On the game lobby webpage Patricia will have a “player profile” button. Once she clicks the button, she would be presented with a pop-up window showing her players statistics composed of the number of games she played, the games she won and lost, her accumulated point, award badges, remaining power charms and game ranking based on her points. Patricia would also see her user id and customized avatar.

8. **Maintain the server:** Hugo a system manager would be required to do a system maintenance once a month. He would schedule the maintenance based in user’s

utilization of the system. After Hugo identifies the times where system utilization is below 3% he will schedule a systemwide maintenance. Tina, a player, would be notified via pop up message next time she logs in about the maintenance. During the maintenance no user would be able to log in or play the game. System downtime should not exceed more than two hours and Hugo should schedule it at least 3 weeks in advance. Hugo would be able to maintain the server through the back-end software that he can accesses through the web portal with his username and password.

9. **Update player interface:** Enrique, a system manager, would be able to update the game interface through the back-end software. He will log in the system through the web portal with his username and password. Enrique would then change layouts, colors, graphics and the game UI according to developers' specifications. Once the updates are completed Hugo will click on a button to update the server. There should not be system downtime for these types of updates. After Hugo completes the updates, they are pushed to the front-end user interface. The next time Tina, a player, logs into the system she would be presented with the new updated UI.

Stakeholders

The Client

The client is our parent company Board Games 2.0 INC (refers as the company). The company is privately own with a market capitalization of 10 million dollars and around 250 employees. The company's main goal is to develop new games based on old board games augmenting their capabilities with new rules, graphics, gamer interaction and award systems. The company has had great success with 4D chess and 4D checkers growing their business and market capitalization by 100% within the past 5 years. The company develops and releases new games in a two-year schedule. 3D tic-tac-toe is one of the projects for the 2020-2022 development run.

The Customer

3D tic-tac-toe is intended for the online gamers market. In 2010 there were around 1 billion gamers worldwide and in 2020 there are around 4.5 billion. This vast market is composed of persons ranging from the ages of 15 to 54. This year revenues projection for the industry are around $196 billion. Demand for online board games has increased 10-fold in the past 5 years. These customers are looking for new and traditional games that have been modified to play online. The most popular and successful games are those that provide the gamers community with competitions and interactions among players as well as rewards and tournaments systems. 3D tic-tac-toe will have all these features and would be updated according to new game trends and technologies.

Hands-On Users of the Product

The main users of the product are our current game platform subscribers and the online gamer community at large. Their main responsibility is to play the game at least once a week, participate in tournaments and interact with other players. These actors must be journeyman of online gaming, be in the 10 to 55 years old bracket and be novice of traditional tic-tac-toe.

Maintenance Users and Service Technicians

System operators and managers will be in charge of deploying the game in the different platforms, of maintaining and updating the system. They are in-house actors with a deep and extensive understanding of the system front and back end (full stack developers) They have worked in close contact with the developing team to understand de game and its functionality. They must have experience in games and product software development.

Other Stakeholders

Sponsor: The company will be sponsoring and promoting the product on their other game platforms.

Testers: There would be a beta testing phase were current users of our other game products will be invited to try and test the game as well as to give feedback.

Business analysts: Our in-house business analyst and game researchers will be providing support and feedback on our competitors and current products success.

Marketing experts: Our in-house marketing will also use the product to advertise new games and product releases.

Legal experts: Our in-house legal team will provide support on the privacy laws and legal framework in the countries where the product will be released.

User Participation

Our research and business analyst team will provide the developers with inside on our current product successes and failures, what has work vs. what has not. They will also provide the knowledge of the latest trends in game features and player satisfaction. They will provide answers to our questions, what do players look for during gameplay? what kind of interactions? This input will help the development team in their UI and features design.

Current users of our game platforms will provide through our beta releases knowledge of interface prototyping and as well as usability requirements. They will be required to play the game for at least 20 hours.

Priorities Assigned to Users

We identified the following priorities in our users,

- Key users: Current company game platform users and subscribers. Currently our platform has 5.1 million players worldwide.
- Secondary users: The online gamers community, system owners and managers.
- Unimportant users: Any user that is not familiar with online gaming. Users that play the game once or have no knowledge of how to play tic-tac-toe.

Mandated Constraints

These are the non-negotiable necessities of the project in question.

Solution Constraints

Description: The product shall be available on the Google Play Store, Apple App Store, as well as a website

Rationale: There are different cohorts of people in all three of those categories and we want this available to as many people as possible so we will target all three audiences.

Fit Rationale: Consistent design between the two applications on the respective mobile devices, and the same theme applied to the Website. Performance should be similar regardless of the device.

Description: The product shall have full cross-platform capabilities. No matter what device the user is using, they should be able to communicate and interact with other players.

Rationale: One of the most enticing features is to play with friends, so different devices should not be the restricting factor for friends to play with each other

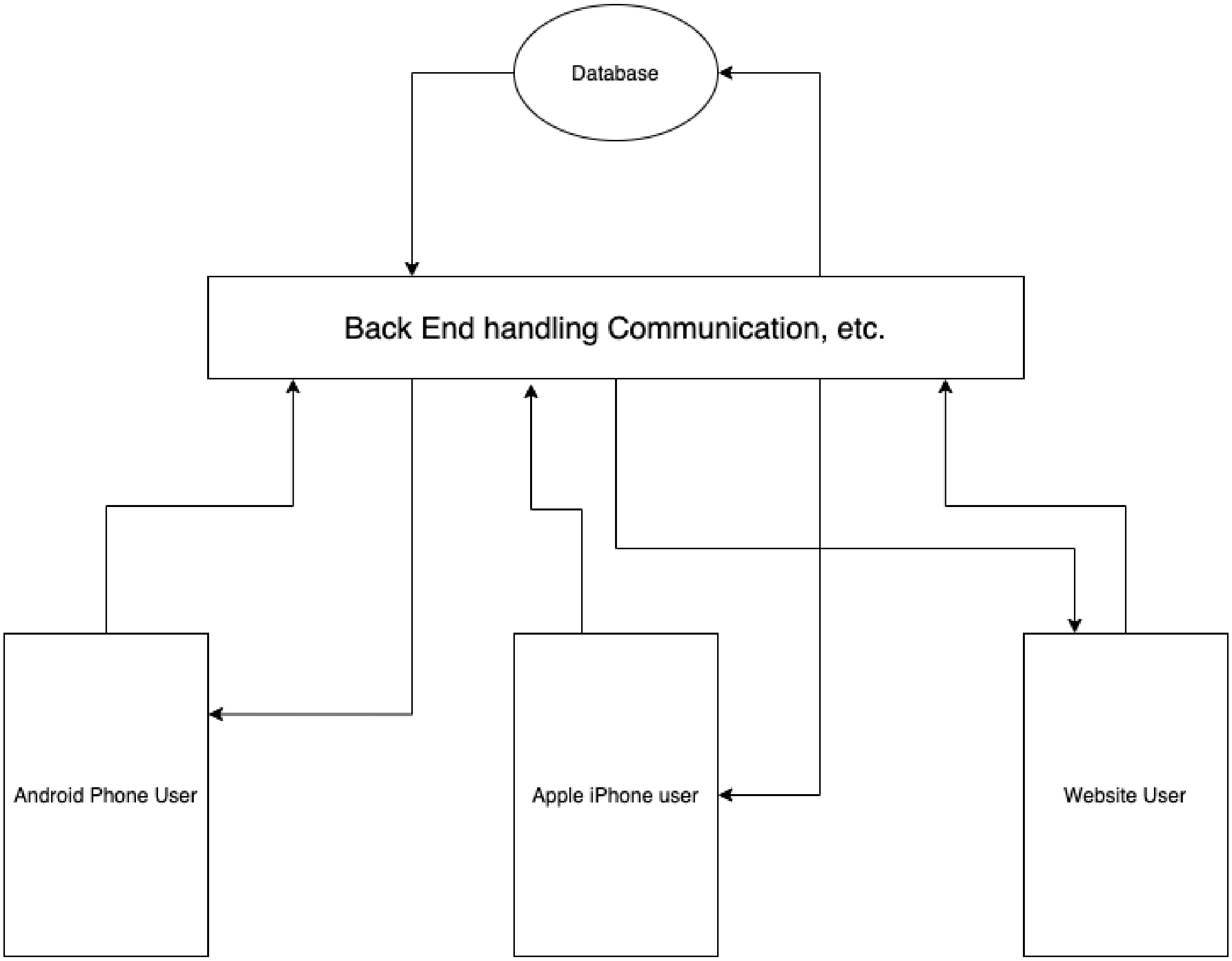
Fit Rationale: Fast connectivity between the three options

Description: The product shall have full cross-platform capabilities. No matter what device the user is using, they should be able to communicate and interact with other players.

Rationale: One of the most enticing features is to play with friends, so different devices should not be the restricting factor for friends to play with each other

Fit Rationale: Fast connectivity between the three options needs to be the standard, it is unlikely that they will keep on coming back if they have connection issues.

Implementation Environment of the Current System



Users on the different platforms are allowed to communicate with each other and play games together. This requires all of the respective Hardwares to be able to access a common server and share data between each other.

Environment of the current System

Partner or Collaborative Applications

- Option to export game data to a Microsoft Excel sheet which will show improvement by means of a graph. This allows the user to have access to their statistics and do what they like with it

- Option to share information with the user’s available applications

  - Share scores on social media

  - Automated text messages to invite other people to play

- Email blast service to keep data organized and send account updates/features update/policy changes
- Work with 4D Chess and 4D Checkers to promote 3D Tic Tac Toe

Off-the-Shelf Software

Google Play Store – It is a necessity for any business creating an application to publish on the Google Play store to give easy access to all the Android users.

Apple App Store – It is a necessity for any business creating an application to publish on the Apple App store to give easy access to all the iPhone users.

AWS Servers – Maintaining user data in a safe place will require the user of Amazon Web Services. They will be retrieved and kept safe through Amazon's services.

Anticipated Workplace Environment

- The messaging of other players should not interfere with the gameplay of a user.
- Any reminders by email should not be frequent such that users think they are spam.
- The product must have simple big and easy directions to help the user navigate
- The product must start off easy to encourage the user to get better at the game
- The product must be visually appealing as that is involved in customer retention

Schedule Constraints

- Window of opportunity: (to take into account)
    - With Corona forcing everybody to spend unprecedented amounts of time in front of a screen, this opportunity needs to be availed. However, the product that is released needs to be top notch and high quality to match the Client's needs
- Deadlines:
    - Initial marketing start: Jan 2021
    - Program sent to testers for functionality: Sept. 2021
    - Hype Buildup/People reviewing the game: Feb 2022
    - Release: Jun 2022

- Notes:
    - Important to release in a Summer since the younger demographic will be off from school and have a lot of free time.
    - Need to create a “Hype Train” with marketing, working with influencers, etc. before final release.

Budget Constraints

Total Budget: $300,000

- Top notch Artificial Intelligent robot. Will need to create a robot based on various models that understands the game and can play it at varying difficulty levels.
- Graphic Designing for logos, and animations for applications/website
- Android and iOS programmers respectively
- Website programmer and designer
- Maintaining server and databases

Naming Conventions and Definitions

Definitions of Key Terms

Users/players are use interchangeable in use cases and they refer to any person that plays the game.

System owners are persons that maintain or update the system from the back-end server.

All terms are used in this document:

Avatar: A player’s customizable character

Rank: The position in hierarchy based on a players wins

Points: Rewarded by completing certain milestones during games

Badges: Rewards granted after reaching a certain number of points

Power-charms: Special abilities a player can use during the game to get an advantage over other competitors

Google Accounts: Any account created through the Gmail service by the user which they have the option to connect to their User account on our platform.

Facebook Accounts: Any account created through the Facebook service by the user which they have the option to connect to their User account on our platform.

Google Play Store: The App store that the application will be published on for users that have Android Phones.

Apple App Store: The App store that the application will be published on for users that have iPhones.

AWS Servers: Servers to hold our data

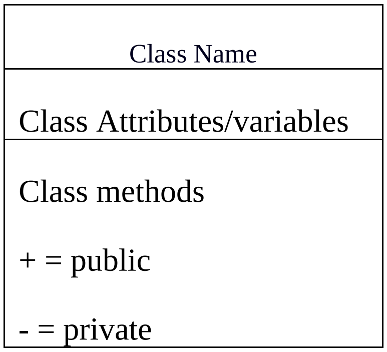
Google Play Games: The social application provided by Google, giving users the ability to compare scores with their friends and get accomplishments.

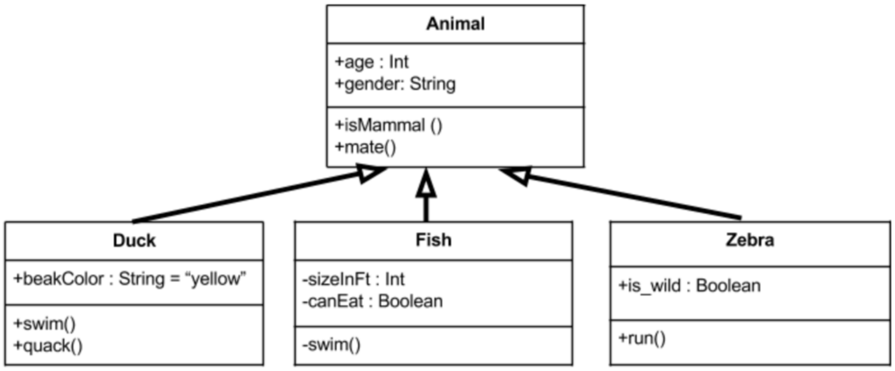
Apple Game Center: The social application provided by Apple, giving users the ability to compare scores with their friends and get accomplishments.

Gameserve: The management and maintenance software use for the back end of 3D Tic-tac-toe.

UML and Other Notation Used in This Document

We will be using the standard UML notations as described in UML Distilled by Martin Fowler. An example UML will be shown below via the Medium article where all basic UML class diagrams elements will be present.





Data Dictionary for Any Included Models

Database: Table to store the perks and power-charms that are available in the game. The database will also have the user requests when they perform a certain move in the game.

Cloud: The cloud will be where vital information of the user will reside. This includes usernames/email, password, guest account, and in game purchases. We will also store the information of the latest clothes on the avatar and their gaming record of wins, losses, etc.

Relevant Facts and Assumptions

Facts

Requirements:

The user must agree to the terms and services which will clearly let the user know what type of data is collected. The data collected will not have their personal data except their name, email, and password. We will collect the data that are required to run the game. For example, score, wins, in game purchases, and their rank.

Audience:

There are around 4.5 billion gamers in the world with that number steadily increasing at around 5.9% each year.

Game Economy:

The global market of online gaming is more than 165 million dollars and will reach around 300 million dollars by 2025. Globally, 44 percent of gamers reported they would subscribe to a service, with gamers ages 18-45, aspiring professionals and expert gamers the most interested. Too high pricing and poor performance are the top obstacles to subscribing. Our game is catered towards young audience as well so making it free-to-play with buy ins in the game is more beneficial for the earnings and profits our investors will get.

Assumptions

We are hoping to work on a subscription-based model of the game at the same time which will have everything from the free-to-play model and some more additional features for the enthusiasts. The game will be primarily in English with future works on different languages. We expect the game to be played on Android 5.0 and iOS 13 at the least. 4G, 5G, and Wi-Fi compatibility will be required to connect to our servers to play the game.

Requirements

Product Use Cases

Use Case Diagrams

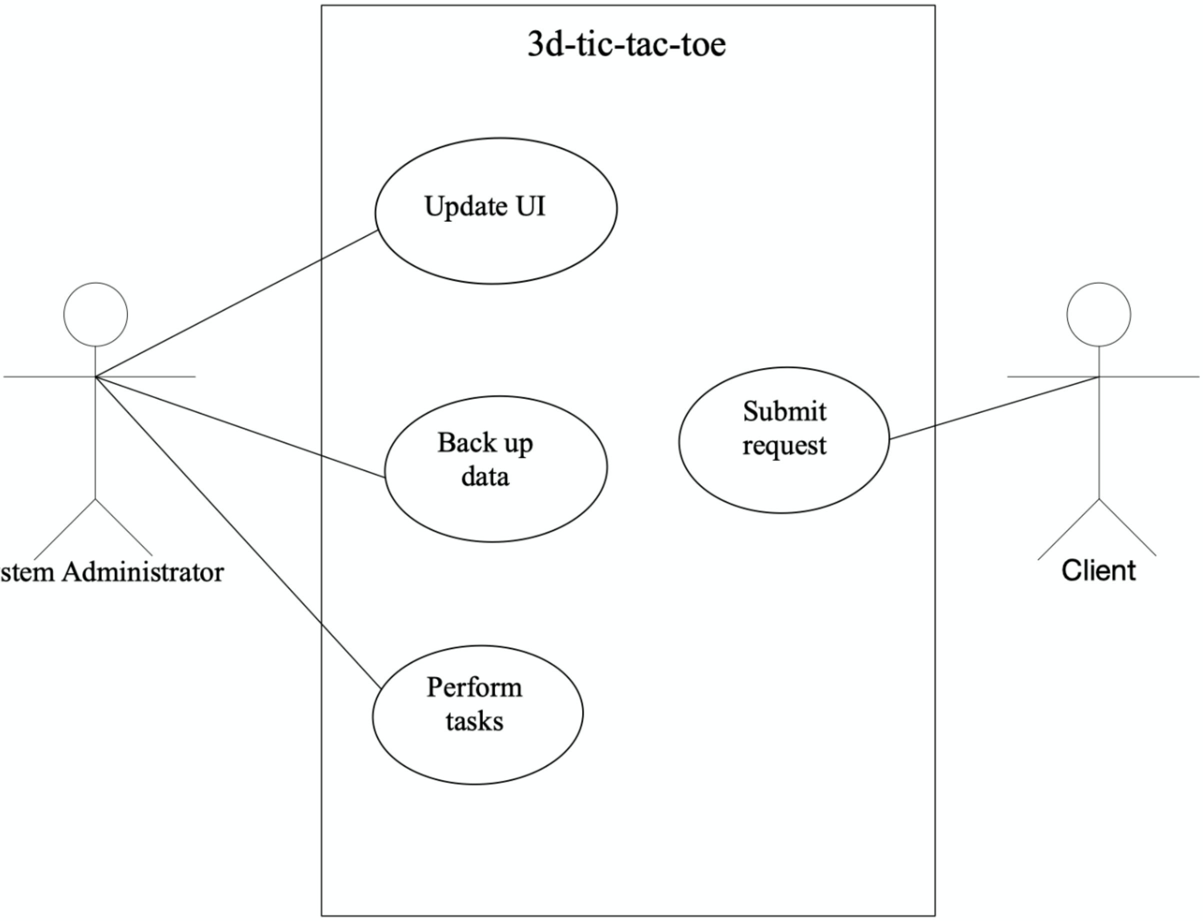


Figure 9.1 Maintenance use case diagram

Product Use Case List

Maintenance
1. Submit maintenance request 2. Backup Data 3. Update UI 4. Perform maintenance tasks

Individual Product Use Cases

Use case ID: maintenace1 Name: Submit maintenance request

pre-conditions: Actor must be login into the system management software “Gameserve”

post-conditions: Actor will have a maintenance request confirmation number

Initiated by: Client

Triggering Event: Client presses the “request service maintenance button” on Gameserve

Additional Actors: System Administrator

Sequence of Events:

1. Client requests system maintenance on Gameserve
    2. System responds by providing the client with a maintenance summary form.
3. Client fills the form with the current maintenance issue and his personal details.
4. Client submits the maintenance form.
    5. System conducts data validation.
    6. Systems displays submitted form and requests confirmation.
7. Client accepts confirmation
    8. System provides client with a confirmation number and email contact.
    9. Systems sends submitted request to the system administrators email inbox.

Alternatives: Request can be done on paper if system fails.

Exceptions: Data validation fails

Use case ID: maintenenace2 Name: Backup data

pre-conditions: System must be in maintenance mode. Data must be refreshed. Actor must be login into Gameserve and have administrator privileges

post-conditions: latest user accounts data would be stored in the backup server. Actor will have a confirmation number.

Initiated by: System administrator

Triggering Event: administrator presses the Backup option on Gameserve main menu.

Additional Actors: None

Sequence of Events:

1. Actor chooses backup option.
    2. System displays menu with backup options.
3. Actor chooses backup data option from menu.
    4. System performs a backup configuration test.
    5. System displays tests performed and passed.
    6. System requests confirmation from actor.
7. Actor provides backup confirmation.
    8. Systems performs backup and displays progress.
    9. Systems provides backup confirmation and an associated number.

Alternatives: Perform on schedule

Exceptions: At least one backup configuration test fails

Use case ID: maintenace3 Name: Update UI

pre-conditions: Actor must have new UI layout file. Actor must be login into Gameserve and have administrator privileges. System must be in maintenance mode. Actor has “update game configurations menu” open.

post-conditions: Game UI has changed and is updated with new file. Actor will have confirmation number.

Initiated by: System administrator

Triggering Event: Administrator chooses update game layout from menu.

Additional Actors: None

Sequence of Events:

10. Administrator chooses update UI from menu.
    11. System performs update system check.
    12. System responds by asking the actor to choose the location of the file.
13. Administrator provides location of file.
    14. Systems validates file.
    15. System performs update and displays its progress.
    16. System provides update confirmation and corresponding number

Alternatives: Perform on schedule

Exceptions: Update or system check fails. Wrong file.

Use case ID: maintenance4 Name: Perform maintenance tasks

pre-conditions: Maintenance must be scheduled for current period. System must be in maintenance mode. Actor must be login into Gameserve and have administrator privileges. Actor must be in the “perform system maintenance” menu option.

post-conditions: The maintenance record file will be updated. Actor will have confirmation number.

Initiated by: System Administrator

Triggering Event: Actor selects perform system maintenance in Gameserve main menu.

Additional Actors: None

Sequence of Events:

17. Actor selects perform system maintenance in main menu.

    18. System responds with a menu with all the available maintenance tasks.

19. Actors select maintenance task/s to perform.

    20. System responds with a list of maintenance task/s to perform.

    21. System displays confirmation message.

22. Actors provides confirmation.

    23. Systems responds by performing requested task/s.

    24. System displays progress.

    25. System updates the maintenance record file.

    26. System provides confirmation and associated number.

Alternatives: Perform on schedule.

Exceptions: At least one maintenance task fails.

Product Use Case List

Pre-Game
1. User Changes Avatars
2. User Chooses Game Mode
3. User Invites friends to a personal lobby
4. User queues with friends
5. User reports an unfriendly user

Use case ID: PreGame1 Name: User Changes Avatars

pre-conditions: The game should open up to show the players' character and should have an option to customize it on the main screen before jumping into a game.

post-conditions: Game UI has changed and is updated with the new avatar. The player will now see the new avatar on the screen.

Initiated by: user

Triggering Event: Administrator chooses update avatar from menu.

Additional Actors: None

Sequence of Events:

1. User chooses to update the avatar from the menu.
2. System performs update avatar check.
3. System responds by asking the actor to choose the location of the file.
4. Administrator provides location of file.
5. Systems validates files.
6. System performs updates and displays its progress.
7. System provides update confirmation and corresponding number.

Alternatives: N/A

Exceptions: Update or system check fails. Wrong file.

Use case ID: PreGame2 Name: User Chooses Game Mode

pre-conditions: The game should show game options on the game screen for the player to decide the preferred game mode to play.

post-conditions: The player is then connected to the appropriate server for that chosen game mode.

Initiated by: user

Triggering Event: Administrator chooses game mode from game screen menu.

Additional Actors: None

Sequence of Events:

1. User chooses to select the game mode from the menu.

2. System performs update game mode check.

3. System responds by asking the actor to choose the location of the file.

4. Administrator provides location of file.

5. Systems validates files.

6. System performs updates and displays its progress.

7. System provides update confirmation and connects the player to the corresponding active server for that game mode.

Alternatives: There is no active server for that game mode and a new server needs to be activated.

Exceptions: Update or system check fails. Connection to the server times out.

Use case ID: PreGame3 Name: User Invites friends to a personal lobby

pre-conditions: The game should allow each player to invite their friends to their server to play among themselves.

post-conditions: A personal game between the friends is set up and all of the users are connected to the same server.

Initiated by: user

Triggering Event: Administrator connects all the users to the same server.

Additional Actors: None

Sequence of Events:

1. Users choose to invite their friends.

2. System responds by asking the actor to retrieve the username for the player(s) that are invited.

3. Administrator provides username(s).

4. Systems validates username(s).

5. System connects the invited player to the player who invited..

6. System then connects the players to an active server.

Alternatives: The invited player declines the request.

Exceptions: No Active servers. The connection times out.

Use case ID: PreGame4 Name: User queues with friends

pre-conditions: The users queue as a party for the game.

post-conditions: The party finds an active server and connects to it.

Initiated by: users

Triggering Event: Administrator looks for an open server with space for players to connect them to it.

Additional Actors: None

Sequence of Events:

1. Administrator queues the players as a party.
2. System responds by requesting the usernames of the players queueing.
3. Administrator provides usernames.
4. Systems validates usernames.
5. System searches for an open server with space for players.
6. System connects the party to an open server with other players.

Alternatives: No open server so the players are connected to a personal server with no extra players.

Exceptions: No Active servers. The connection times out.

Use case ID: PreGame5 Name: User reports an unfriendly user

pre-conditions: The game should have an option for users to report an unfriendly user for toxicity or harassment.

post-conditions: The reported player is then investigated for their behavior.

Initiated by: user

Triggering Event: Administrator writes a report for the reported player.

Additional Actors: None

Sequence of Events:

1. User chooses a player for their behavior.

2. System responds by asking the username of the user that reported and the user who is reported.

3. Administrator provides the usernames.

4. Systems validates usernames.

5. System writes a report for the incident case.

6. System provides the report to the legal admin in charge of harassment control.

Alternatives: N/A

Exceptions: Update or system check fails. Wrong username.

Product Use Case List

Gameplay
1. User starts game 2. User achieves milestone

3. User sends message
4. User Uses powerup
5. User wins

Individual Product Use Cases

Use case ID: gameplay1 Name: User starts game

pre-conditions: Must be connected to the internet and have correct game mode chosen

post-conditions: The user has started the game with friends

Initiated by: The user

Triggering Event: The user queues up game mode and presses ‘Play’

Additional Actors: The system and the database where the user accounts will be stored.

Sequence of Events:

1. The user opens the game.
2. The system asks if the user wants to invite friends.
    - The user could play in public lobby as well
3. The user chooses to queue up game and press ‘play’.
4. The user enters a loading screen where the system sets up game mode with friends or with other players in a public lobby.
5. The system checks the database to check that the credentials are correct.
6. The user has entered the game.

Alternatives: N/A

Exceptions: The user has internet issues and cannot start game

Use case ID: gameplay2 Name: User Achieves Milestone

pre-conditions: User is in the game and has met the requirement for a certain milestone.

post-conditions: Milestone award is bestowed to user.

Initiated by: The system

Triggering Event: The user completes a designated task that is pre-set by the game system

Additional Actors: The database where the milestone requirements will be stored.

Sequence of Events:

1. The user is playing the game.
2. The user has completed a certain challenging task.
3. The system runs a test case to ensure the milestone task is met.
4. If yes, the system displays a textbox with ‘Congratulations, you’ve achieved a milestone!’.

Alternatives: N/A

Exceptions: N/A

Use case ID: gameplay3 Name: User Sends Message

pre-conditions: User is in a game mode and has selected the chat box.

post-conditions: The user sent message/emoji to other players

Initiated by: The user

Triggering Event: The user clicks on chat box during a game

Additional Actors: The system and the database where the emoji options will be stored

Sequence of Events:

1. The user clicks on the chat box.
2. The system opens chat box and provides an option for text message or emoji.
3. The user selects chat option
    - Or user selects emoji option
4. The system sends message from server to client for other players to receive the message.

Alternatives: N/A

Exceptions: N/A

Use case ID: gameplay4 Name: User uses power charm

pre-conditions: User has enough points to initiate power charm ability

post-conditions: The user uses power charm ability

Initiated by: The user

Triggering Event: Earning points and/or use case, gameplay2

Additional Actors: N/A

Sequence of Events:

1. The system displays all the information about the points the player has and when they can use a certain power-charm move.
2. The user plays game and earns points as they progress through the game.
3. The user earns enough points to use a power charm.
4. The use selects the Power charm option and selects the power charm they would like to use.
5. System communicates with database to initiate the power charm.
6. The power charm animation takes over the display of the users in the game.
7. The power charm is complete, and the users continue to play the game.

Alternatives: User chooses alternate power-charm

Exceptions: A separate user initiates a power-charm at the same time that blocks out the first user's power charm

Use case ID: gameplay5 Name: User wins game

pre-conditions: User has connected 3 X/O's on all 3 dimensions on the cube game layout

post-conditions: The user wins and game ends Initiated by: The user Triggering Event: Connecting 3 X/O’s on all 3 dimensions on the cube game layout Additional Actors: N/A
Sequence of Events: 1. The user connects 3 X/O’s on all 3 dimensions on the cube game layout. 2. The system checks to make sure the player has met the winning criteria. 3. The system displays a winning animation for the users in the game lobby. 4. The system awards points to each player based on their game performance. 5. All users exit the game mode and return to the main lobby.
Alternatives: N/A Exceptions: A power charm is used prior to user winning.

Product Use Case List

Postgame
1. Update scoreboard 2. Share game summary 3. Chat with other players 4. Request for playing again 5. Quit the game

Use Case ID: PostGame1	Name: Update Scoreboard
Pre-conditions: Completion of Game	
Post-conditions: Display of Scoreboard	
Initiated by: Player	
Triggering Event: Last move completed by a specific player	

Additional Actors: Other Players

Sequence of Events:

1. Player makes the last move
    1. System concludes that the game has been completed and there is a clear winner based on the points earned and the current board
    2. System posts board for all the players to see
2. Players see complete game statistics regarding the game played
    1. System presents options to play again, quit, or share
3. Players make their respective decision.

Alternatives: Potential tie, System needs to be prepared for that possibility

Exceptions: Player leaving last second, System needs to ensure that that does not interfere with the display for other players

Use Case ID: PostGame2 **Name:** Share game summary

Pre-conditions: Game must be completed

Post-conditions: Game summary shared on the social media platform of the players choice

Initiated by: Player

Triggering Event: Player selecting option to share game summary

Additional Actors: System, Usage of API from Social Media sites

Sequence of Events:

1. Player’s game end
    1. System presents the various next actions along with the scoreboard
2. Player selects share option for specific social media
        1. System opens up pop-up for the respective social media
        2. If not logged in, allows the user to login. Otherwise:
3. User types message and subsequently shares on aforementioned social media

    1. System returns user back to game with the options to play again, quit, or share

Alternatives: N/A

Exceptions: Connection between System and Social Media fails, System needs to inform user and return to menu

Use Case ID: PostGame3 **Name:** Chatting with other players

Pre-conditions: Game ended already

Post-conditions: Ideas communicated between the players

Initiated by: Various players

Triggering Event: Players communicating via voice or text using the System’s features

Additional Actors: System

Sequence of Events:

1. Game is completed
    1. System presents scoreboard
    2. System ensures that the players are still able to communicate with each other since they are in the same lobby.
2. Players can discuss how the game went
    1. System allows for communication to continue
    2. System also has the various options available for the players

Alternatives: Only text or only voice communication for a player that requested

Exceptions: Microphone of a user not working when they are attempting to communicate, mention that to them.

Use Case ID: PostGame4 **Name:** Request for playing again

Pre-conditions: Game is completed

Post-conditions: Another game starts

Initiated by: Player

Triggering Event: Player selects option to play again

Additional Actors: Other players, System

Sequence of Events:

1. Game is completed
    1. System presents scoreboard
2. Player selects option to replay
    1. System enters Player’s avatar in a lobby, waiting for other players
    2. Once System has enough players to start a game
3. Main Host has option to start game or wait for more players
    1. System starts game once told to

Alternatives: Player does not choose to play again, Player choose to play then changes mind

Exceptions: If player starts another game but leaves, an AI will play in his stead.

Use Case ID: PostGame5 **Name:** Quit the game

Pre-conditions: Game ends

Post-conditions: Player has left this specific game

Initiated by: Player

Triggering Event: Player selects option to quit

Additional Actors: System

Sequence of Events:

1. Game is completed

    1. System presents scoreboard
2. Player selects option to quit game
    1. System takes player back to main menu, presenting options to go into another lobby
3. Player either leaves the game entirely or joins another lobby

Alternatives: Player abruptly exits by closing tab or application

Exceptions: Player stays in

Functional Requirements

Requirement #: M1

Requirement Type: Functional

Event/use case#: maintenace1

Description: The system shall provide the user with the means to enter data.

Rationale: The user needs to provide information about an action he wants done.

Originator: Client

Fit Criterion: The request is received by the appropriate actors and matches the provided data.

Customers Satisfaction: 4

Customer Dissatisfaction: 5

Priority: high

History: Created 10/23/20

Requirement #: M2

Requirement Type: Functional

Event/use case#: maintenance1, maintenance3, maintenance4

Description: The system shall provide a means for the user to confirm an action.

Rationale: To get confirmation from the user on an action the system will perform

Originator: Client, system administrator

Fit Criterion: The system executes de action the user confirmed.

Customers Satisfaction: 4

Customer Dissatisfaction: 5

Priority: high

History: Created 10/23/20

Requirement #: M3
Requirement Type: Functional
Event/use case#: maintenace1

Description: The system shall be able to route messages among users.
Rationale: System users need to communicate their requests.
Originator: Client, System administrator
Fit Criterion: message N sent by user1 to user N, is the same message N user N received from user1.
Customers Satisfaction: 4
Customer Dissatisfaction: 5
Priority: high
History: Created 10/23/20

Requirement #: M4
Requirement Type: Functional
Event/use case#: maintenance1, maintenance4

Description: The system shall provide feedback to the user about his input.
Rationale: The user needs to know if the data he inputs into the system is correct.
Originator: Client, system administrator
Fit Criterion: The data feedback provided by the system matches the data the user provided

Customers Satisfaction: 4
Customer Dissatisfaction: 5
Priority: high
History: Created 10/23/20

Requirement #: M5
Requirement Type: Functional
Event/use case#: maintenace2, maintenance3, maintenance4

Description: The system shall provide the user with feedback on performed tasks and their progress.
Rationale: The user needs to know what tasks the system is performing and their progress.
Originator: Client
Fit Criterion: The user is aware what task/s the system is performing and their progress.

Customers Satisfaction: 4
Customer Dissatisfaction: 5
Priority: Medium
History: Created 10/23/20

Requirement #: M6
Requirement Type: Functional
Event/use case#: maintenace2

Description: The system shall be able to write its data contents into the backup server.
Rationale: The systems needs to be able to recoup from loss or corrupted data.
Originator: System Administrator
Fit Criterion: Current system data matches data on backup server.

Customers Satisfaction: 5
Customer Dissatisfaction: 5
Priority: High
History: Created 10/23/20

Requirement #: M7
Requirement Type: Functional
Event/use case#: maintenace3

Description: The system shall be able to read files provided by the user.
Rationale: The user needs to maintain and update the system.
Originator: System Administrator
Fit Criterion: System is able to read and process the file format provided by the user.

Customers Satisfaction: 5
Customer Dissatisfaction: 5
Priority: High
History: Created 10/23/20

Requirement #: PG1
Requirement Type: Functional
Event/use case#: PreGame1
Description: The system shall have a mechanism to allow the user to change avatars
Rationale: The user should be allowed to change their avatar.
Originator: Client, system administrator
Fit Criterion: Ensure that any user has the ability to change the avatar to their preferred. It could be an avatar they earned or bought.
Customers Satisfaction: 4
Customer Dissatisfaction: 5
Priority: high
History: Created 10/31/20

Requirement #: PG2
Requirement Type: Functional
Event/use case#: PreGame3, PostGame3, Gameplay3
Description: The system must allow for in-game communication during the party server
Rationale: Data sent by one user should be visible to everyone else in the party
Originator: Client, system administrator
Fit Criterion: Data entered by the user matches harassment policy.
Customers Satisfaction: 4
Customer Dissatisfaction: 5
Priority: high
History: Created 10/23/20

Requirement #: PG3
Requirement Type: Functional
Event/use case#: PreGame5
Description: The system must have a mechanism for users to report other unfriendly players.
Rationale: Multiple options must be presented so the user can decide what type of harassment was done.
Originator: user, system administrator
Fit Criterion: Users have full ability to send as many reports as they like.
Customers Satisfaction: 5
Customer Dissatisfaction: 5
Priority: high
History: Created 10/23/20

Requirement #: GP1
Requirement Type: Functional
Event/use case#: gameplay1

Description: The system shall be let the user know once a session has begun
Rationale: Once the user initiates a game, the game start. Thus the System must have a way of mentioning that end to the user
Originator: System administrator
Fit Criterion: To test this, all possible ways of starting a game must be covered for and must lead to the same result

Customers Satisfaction: 4
Customer Dissatisfaction: 5
Priority: high
History: Created 10/23/20

Requirement #: GP2
Requirement Type: Functional
Event/use case#: gameplay2

Description: The system shall be let the user know once a milestone is achieved
Rationale: During the game, the user can use certain power charm moves after attaining certain amount of milestones and achievements. This will help the user be motivated to buy new items and continue to play the game to strive for a new item.
Originator: System administrator
Fit Criterion: To test this, all possible ways of earning a milestone must be covered for and must lead to the same result

Customers Satisfaction: 5
Customer Dissatisfaction: 5
Priority: high
History: Created 10/23/20

Requirement #: GP3
Requirement Type: Functional
Event/use case#: gameplay4

Description: The system shall be let the user know when a power charm feature is being used
Rationale: Once a user initiates his power charm, the other users in the game lobby will need to strategically plan their next moves to compete for a win
Originator: System administrator
Fit Criterion: To test this, all possible ways of initiating a power charm feature must be covered for and must lead to the same result

Customers Satisfaction: 5
Customer Dissatisfaction: 5
Priority: high
History: Created 10/23/20

Requirement #: PG1
Requirement Type: Functional
Event/use case#: PostGame1, gameplay5

Description: The system shall be let the user know once a user has won and the session has been completed
Rationale: Once a game starts, it must end. Thus the System must have a way of mentioning that end to the user
Originator: System administrator

Fit Criterion: To test this, all possible ways of ending a game must be covered for and must lead to the same result

Customers Satisfaction: 4
Customer Dissatisfaction: 5
Priority: high
History: Created 10/23/20

Requirement #: PG2
Requirement Type: Functional
Event/use case#: PostGame2

Description: The system shall have a mechanism to allow the user to share their scores
Rationale: The user should be allowed to share their results with their friends on all the major Social Media platforms
Originator: Client, system administrator
Fit Criterion: Ensure that any user has the ability to connect to a valid account on other websites and successfully make a post of some kind on it.

Customers Satisfaction: 4
Customer Dissatisfaction: 5
Priority: high
History: Created 10/23/20

Requirement #: PG3
Requirement Type: Functional
Event/use case#: PostGame3, gameplay3

Description: The system must allow for in-game communication
Rationale: Data entered by the user must be consistent
Originator: Client, system administrator
Fit Criterion: Data entered by the user matches data specifications

Customers Satisfaction: 4
Customer Dissatisfaction: 5
Priority: high
History: Created 10/23/20

Requirement #: PG4
Requirement Type: Functional
Event/use case#: PostGame4, PostGame5

Description: The system must have a mechanism for users to choose an option
Rationale: Multiple options must be presented so the user can decide
Originator: user, system administrator
Fit Criterion: User has full ability to make any decision they like

Customers Satisfaction: 5
Customer Dissatisfaction: 5
Priority: high
History: Created 10/23/20

Data Requirements

ID# - DT1

Description: Game should be able to know the amount of currency a player possesses.

Rationale: A player should be able to know this data from the game so that he can buy units, items, etc.

Fit Criterion: N/A

Acceptance Tests: TDT1

ID# - DT2

Description: Game should contain a user's play history.

Rationale: A player should be able to know this data from the game so that he can see stats and improve on future games.

Fit Criterion: If it can accurately count the currency overtime this requirement should be met.

Acceptance Tests: TDT2

Performance Requirements

Speed and Latency Requirements

Requirement # P1

Description: Interactions between users and the product may have a maximum lag of 2 seconds.

Rationale: Any interaction that goes from the user to the product to another user may have the aforementioned maximum lag. Anything beyond that will not be acceptable since the users should not feel like they are waiting on the product.

Fit Criterion: N/A (Above)

Acceptance Tests: TP1

Requirement # P2

Description: Any text interaction strictly between two users may have a maximum lag of ¾ second.

Rationale: If users wish to communicate via text using the product, they must be able to in a near-instant fashion.

Fit Criterion: N/A (Above)

Acceptance Tests: TP1

Requirement # P3

Description: Any interaction between the product and any outside source may have a maximum 3 second lag from the product's side

Rationale: Since connecting with other sources upon the user's request may take a bit more time, need to accommodate for that.

Fit Criterion: N/A (Above)

Acceptance Tests: TP1

Precision or Accuracy Requirements

Requirement # P4

Description: All monetary transactions will be accurate to two decimal places where applicable. Where not applicable, it will be accurate to the one's place or no place's.

Rationale: Any monetary-related services must be as accurate as possible since the user should not feel like they are being scammed.

Fit Criterion: To test, all currencies throughout the world will be cross-referenced.

Acceptance Tests: TP2

Requirement # P5

Description: All services related to time in the product may be accurate up to seconds.

Rationale: Users should know the seconds remaining on any time-sensitive function of the product so that they can do what they feel is best for their situation.

Fit Criterion: N/A

Acceptance Tests: TP3

Capacity Requirements

Requirement # P6

Description: The product shall accommodate for a maximum of 5000 users at peak time, from 5:00pm to 10:pm. At other periods there will be a maximum of 1000 users.

Rationale: For the peak times that gamers are on, the product should be able to handle every single person that would like to play. Not doing so may turn people away from the product itself.

Fit Criterion: Listed above

Acceptance Tests: TP4

Requirement # P7

Description: During service times, the product shall accommodate a maximum of 3500 users at peak time and a maximum of 750 at other periods.

Rationale: Some functionality may not be as robust and working fully since there may be a service repair going on, nevertheless there needs to be a specific amount of users that should still be able to play

Fit Criterion: Listed above

Acceptance Tests: TP4

Dependability Requirements

Reliability Requirements

Requirement # D1

Description: The product shall not have a system wide failure except in very specific cases.

Rationale: The product must be available 24/7 to ensure that anybody around the globe that would like to play it can play it. Only exceptions may be an expected shutdown for fixes or security breaches; users must be required to know beforehand

Fit Criterion: To test, the product must be able to go through a shutdown and return back safely in a specified amount of time.

Acceptance Tests: TD1

Requirement # D2

Description: The product shall not lose any data of any user in the event of any failure.

Rationale: User Data is one of the most precious commodities for the product and users should not feel like they have lost progress to any degree, otherwise they might not be inclined to come back.

Fit Criterion: To test, ensure that the data that is safely backed up in a secure location can be used to bring back potential lost data.

Acceptance Tests: TD2

Availability Requirements

Requirement # D3

Description: Any failure of the product, expected or unexpected, may not go on beyond 5 hours

Rationale: Whether the reasons for the shutdown is fixing a bug or a security breach, the problem should be solved within 5 hours. Regular users should not have to wait longer than that to play a game.

Fit Criterion: To test, all teams associated with the shutdown should have well communication between each other about the tasks at hand and the related deadlines.

Acceptance Tests: TM&S2

Requirement # D4

Description: The product shall be available for use by the users for 700/720 hours in a month, to account for any fixes or security breaches.

Rationale: The 20 hours allows for leeway for the developers to accomplish their job and fix any problems that may arise. At the same time, the product will be available for 97% of the time in a month which is a good percentage that needs to be required.

Fit Criterion: N/A

Acceptance Tests: TD3

Robustness or Fault-Tolerance Requirements

Requirement # D5

Description: The product should allow the user to be able to play for at least ten minutes even with connectivity failure, given that the user is playing against a non-user.

Rationale: The user should not be disrupted if they are using the product for as long as possible. This gives them some more time to interact with the product even though they may not necessarily be connected.

Fit Criterion: Ensure that the product has the ability to load up actions on a local level in the case of a network failure.

Acceptance Tests: TD4

Requirement # D6

Description: The product shall have an auto-save function so that the user does not lose any of their progress in the event of a fault, either on the product's side or the user's side.

Rationale: The user should never feel like they will lose all of their progress and accomplishments if something goes awry. Thus, the product needs to account for all of the possibilities.

Fit Criterion: To test, the system should be forcefully shutdown in a variety of scenarios. In every case, the data of the user shall not be corrupted.

Acceptance Tests: TD2

Requirement # D7

Description: The product shall decrease the level of graphics in the event of a server overload with too many users. This will help in the speed aspect of the users the are currently playing.

Rationale: Even if the product is getting the its limits in terms of the data that it can handle, it shall not decrease in the functionality and gameplay of the user. They shall continue playing the game in the same expected speeds, albeit with lower graphics.

Fit Criterion: To test, ensure that the system can automatically switch users to a lower set of graphics without human intervention, once a specific command has been fulfilled.

Acceptance Tests: TD4

Safety-Critical Requirements

This requirement does not apply to our project.

Maintainability and Supportability Requirements

Maintenance Requirements

Requirement #M&S1

Description: The system administrator must complete maintenance requests within 72 hours.

Rationale: Maintenance request need to be implemented to keep the system from entering an erroneous state.

Fit Criterion: Upon receiving a maintenance request the system administrator completes the request within 72 hours

Acceptance Tests: TM&S1

Requirement # M&S2

Description: The hardware technician must be able to add new servers within 48 hours.

Rationale: As the user population grows the system will need more computational and storage capacity.

Fit Criterion: New hardware is installed within 48 hours after receiving the request

Acceptance Tests: TM&S1

Requirement # M&S3

Description: The system cannot be down for more than 5 hours.

Rationale: System needs to be accessed by the users to prevent user dissatisfaction.

Fit Criterion: System operation is restored within 5 hours after going down.

Acceptance Tests: TM&S2

Requirement # M&S4

Description: System maintenance must be performed by a system administrator.

Rationale: Due to the systems complexity only authorized and trained personnel can perform maintenance tasks

Fit Criterion: System maintenance operations are performed by authorized users.

Acceptance Tests: TM&S3

Supportability Requirements

Requirement # M&S5

Description: The system must provide support about its user interface.

Rationale: New users need to know how to interact with the system.

Fit Criterion: User must be able to navigate the system interface after accessing the systems interface tutorial.

Acceptance Tests: TM&S4

Requirement # M&S6

Description: The system must provide help on how to play the game.

Rationale: New users need to know how to play the game.

Fit Criterion: Upon accessing the game help a new user must be able to win the game tutorial match.

Acceptance Tests: TM&S4

Requirement # M&S7

Description: Game and system help must be developed by the development team.

Rationale: Support must be clear, concise and developed by knowledgeable actors.

Fit Criterion: After accessing the game or system support a user will be able to play the game and navigate the UI for at least two times without help.

Acceptance Tests: TM&S4

Adaptability Requirements

Requirement # M&S8

Description: The game must be able to run on Google Chrome, Mozilla Firefox, Opera and Apple Safari.

Rationale: The game must be playable on the most popular web browsers in order to reach the highest number of customers

Fit Criterion: Game operates properly and without errors on the above mention web browsers.

Acceptance Tests: TM&S5

Requirement # M&S9

Description: The game must be able to run on Apple iOS and Android.

Rationale: The game must be playable on the most popular mobile platforms in order to reach the highest number of customers.

Fit Criterion: Game operates properly and without errors on the above mobile platforms.

Acceptance Tests: TM&S6

Requirement # M&S10

Description: The game will be distributed through the Apple App Store and Google Store platforms.

Rationale: The game needs to reach the highest number of customers possible.

Fit Criterion: Game will be available for download on the above mention App stores.

Acceptance Tests: TM&S7

Scalability or Extensibility Requirements

Requirement # M&S11

Description: The system must be able to handle our current number of companywide subscribers.

Rationale: Any of our current customers may want to play the game.

Fit Criterion: Server capacity must be able to handle our current number of server requests.

Acceptance Tests: TM&S8

Requirement # M&S12

Description: The system must be able to process multiple requests from different platforms.

Rationale: All system user must be able to access and play the game

Fit Criterion: Multiple users in different platforms must be able to play at the same time.

Acceptance Tests: TM&S8

Longevity Requirements

Requirement # M&S13

Description: The system should last for 7 years

Rationale: Our current research shows that game usage and life last for about 5 years

Fit Criterion: The system must be operational for 7 years.

Acceptance Tests: TM&S9

Security Requirements

Access Requirements

ID# - S1

Description: Only the developers should be able to see crash reports and logistics. However, users will be notified of crashes.

Rationale: Blocking users from accessing crash reports ensures confidentiality of the application and background processes. Furthermore, it ensures that bugs won't be altered to pose as an attack to the application.

Fit Criterion: General users should not be able to access crash reports.

Acceptance Tests: TS1

ID# - S2

Description: Only developers will have access to the source-code. General users should not have access the code or be able to read or write the source code.

Rationale: Blocking users from accessing the source code is crucial for preventing any hacking and to make sure the application's functionality does not change.

Fit Criterion: General users should not be able to access the source code

Acceptance Tests: TS2

Integrity Requirements

ID# - S3

Description: The system should not allow users to access other users' sensitive information.

Rationale: Keeping the users' sensitive information confidential is crucial as the user could have payment information and personal information as well.

Fit Criterion: General users should not have access to other users' personal information.

Acceptance Tests: TS3

Privacy Requirements

ID# - S4

Description: The application should have a terms and agreement page before the user uses the application that informs users of all privacy standards.

Rationale: Users will feel more comfortable using an application that lists the user's rights regarding their own security and privacy rights.

Fit Criterion: The application should display the terms and agreement page prior to starting the application for the first time and update the user on any changes.

Acceptance Tests: TS4

ID# - S5

Description: Users should be able to terminate their account along with any information stored in that account profile.

Rationale: Users should be treated fairly with their personal information being erased when they choose to delete their profile. This ensures that data will not be leaked or accessed after termination.

Fit Criterion: User should be allowed to delete their account along with the database containing the account should not contain account info after deletion.

Acceptance Tests: TS5

Audit Requirements

ID# - S6

Description: The application should track user data regarding ratings, multiplayer activity, and any update to the database.

Rationale: Keeping and tracking user data will allow developers to improve the application based on the user's activity.

Fit Criterion: Tracking user data will be allowed

Acceptance Tests: TS6

Immunity Requirements

ID# - S7

Description: The system should be hosted by third party security software. These should be constantly backing up and keeping information encrypted.

Rationale: With constant feedback from security firms, the application will be free of any unauthorized packets being set, and any other malware from corrupting files.

Fit Criterion: There will be multiple rounds of tests placed such as OWASP. These tests will have multiple iterations to protect personal data.

Acceptance Tests: TS7

Usability and Humanity Requirements

Ease of Use Requirements

Requirement # U&H1

Description: The game should open up to show the players' character and should have an option to customize it on the main screen before jumping into a game.

Rationale: It is important to have the ease of changing the avatars and customizing it since our game economy is dependent on players buying into the game. We need to make sure it is easy to work

Fit Criterion: It should open the customization screen in the first ten seconds. The customization screen needs to be visible before a game starts

Acceptance Tests: TU&H1

Requirement # U&H2

Description: The game description must have a "how to" button on the main screen for users who want to refresh their game rules

Rationale: Since 3D Tic-Tic-Toe is an extension of a simple historic game, we need to make sure that the rules are set and are easy to understand.

Fit Criterion: A 10-year-old player should be able to understand the rules of the game when first read.

Acceptance Tests: TU&H2

Personalization and Internationalization Requirements

Requirement # U&H3

Description: The game must be available in multiple languages but most importantly in English, Korean, Japanese, Mandarin, Arabic, and the European languages.

Rationale: Since it is going to be an international game, we need to have multiple languages in the game. Especially the "how to" section of the game and settings.

Fit Criterion: It should be at least translated into the top 5 most common languages by release and continues to add languages for our international player base.

Acceptance Tests: TU&H3

Learning Requirements

Requirement # U&H4

Description: When first opening the app and making an account, the user goes through a tutorial in a language of their choice.

Rationale: 3D Tic-Tac-Toe has important features and rules that each player needs to understand to fully enjoy the game.

Fit Criterion: The game should put the new player with 3 bots as a practice game and an option to redo the game until the player feels confident in their skills and understanding of the game.

Acceptance Tests: TU&H4

Understandability and Politeness Requirements

Requirement # U&H5

Description: The game should be easy for a middle schooler.

Rationale: Since most middle schoolers now have a smartphone and are connected to the internet. We need our game to be easy enough for middle schoolers to play.

Fit Criterion: The game will be tested with middle and high schoolers.

Acceptance Tests: TU&H5

Requirement # U&H6

Description: The game will have symbols and buttons on the interactive game screen while playing.

Rationale: Since we will have powers ups and other buttons during the game, we need them to be in a visually pleasing environment for players to understand and utilize while playing.

Fit Criterion: All important buttons, power ups, and timers must be clear to the user when playing.

Acceptance Tests: TU&H6

Accessibility Requirements

Requirement # U&H7

Description: The game must be accessible to players with disabilities.

Rationale: We want our players from every background to have a good experience while playing.

Fit Criterion: We will test the game with players with Special needs and mild autism to make sure we reach every potential player.

Acceptance Tests: TU&H7

User Documentation Requirements

Requirement # U&H8

Description: The developers must put the "terms of use" and "in game harassment policy" documents in the game.

Rationale: Since we want our users to have fun playing the game and take out any possible toxicity and harassment, we will make all of our players sign each policy.

Fit Criterion: The developers must make a "complaint form" for players to report other players for toxic behavior and any kind of harassment. We also need to make sure that harassment policy is updated every 3 months, so 4 times in a year.

Acceptance Tests: TU&H8

Training Requirements

Requirement # U&H9

Description: For training, the user will learn about the app and its features through a step-by-step tutorial.

Rationale: The app has crucial features that would be handled best when the player has learned the app's functions. Otherwise, it would be hard to understand if the user does not follow the tutorial.

Fit Criterion: The user must be trained to be competent enough to use the interactive buttons and are able to have fun.

Acceptance Tests: TU&H9

Look and Feel Requirements

Appearance Requirements

Requirement # LaF1

Description: The product shall adhere to all the branding standards of Board Games 2.0 INC.

Rationale: All of the products must look the same, thus this product must as well.

Fit Criterion: To test, design mockups will first be checked by the design department of Board Games 2.0 INC. to ensure that all standards are being to a certain extent.

Acceptance Tests: TLaF1

Requirement # LaF2

Description: The product shall have different colors to be associated with the rank of a particular user.

Rationale: The different colors will aspire the lower-ranked users to get to the higher rank. It will also establish color schemes that will be changing but consistent for all users.

Fit Criterion: To test, all colors will be cycled through to make sure they are still appealing and do not break any of the client's aforementioned standards.

Acceptance Tests: TLaF1

Requirement # LaF3

Description: The product shall have a bright and inviting appearance

Rationale: The intended audience is the general public and so they should feel like they're wanted when using said product.

Fit Criterion: To test, the product must be cross-referenced with other popular games with a bright color scheme.

Acceptance Tests: TLaF2

Style Requirements

Requirement # LaF4

Description: The product shall incite a willingness in the user to play the game

Rationale: The product shall create a desire in the user that may be mysterious through text, images, etc. which will want the users to try the game out.

Fit Criterion: To test, 65 percent of initial customers shall feel a desire to play the game

Acceptance Tests: TLaF2

Requirement # LaF5

Description: The game shall be encouraging to first time players helping them through the motions

Rationale: The product shall have a low bar for entry and making progress in the beginning paired with encouraging words.

Fit Criterion: To test, a sample of customers will be shown the game and given the option to play or leave the game. 70% of them must at least try to play one time.

Acceptance Tests: TLaF2

Requirement # LaF6

Description: The product shall make the user feel a sense of accomplishment with their wins.

Rationale: The product should allow the user to feel like the problem is easy enough to solve but also hard enough to create that sense of accomplishment inside of them.

Fit Criterion: To test, the initial customers will be surveyed how they feel and 70% of them must feel like they solved something that was non-trivial but doable.

Acceptance Tests: TLaF2

Operational and Environmental Requirements

Expected Physical Environment

Requirement #O&E1

Description: The product should be able to work on any smartphone or desktop environment.

Rationale: The application does require an internet connection for multiplayer play, so it's important that a smartphone or desktop is required .

Fit Criterion: User should be able to play anywhere in their home or where their device is functioning with internet connection .

Acceptance Tests: TO&E1

Requirements for Interfacing with Adjacent Systems

Requirement #O&E2

Description: The application should work with the operating system of different mobile devices/computers as well as be at an efficient ram and data storage.

Rationale: The game should be practical by not taking up too much storage on the user's device. We would like to keep it efficient and compact while not slowing down any performance.

Fit Criterion: The product should work on all mobile devices computers at an efficient rate.

Acceptance Tests: TO&E2

Productization Requirements

Requirement #O&E3

Description: The product shall be distributed free of charge as an application package on the Google and Apple App Stores for download.

Rationale: Product needs to be distributed widely for better market penetration.

Fit Criterion: Game App should would be available on the mobile app stores.

Acceptance Tests: TM&S7

Requirement #O&E4

Description: The product shall be downloadable and installed by the user of any mobile platform without recourse.

Rationale: Product should be easy to download and install.

Fit Criterion: Game App is downloaded to a mobile device from the device app store.

Acceptance Tests: TM&S7

Release Requirements

Requirement #O&E5

Description: An update version of the product shall be released once a year.

Rationale: The product needs to be competitive and up to date with current game trends and technologies.

Fit Criterion: Developers will work on bug fixes and incorporate new features into the updated version. Budget should not exceed 50K per release.

Acceptance Tests: TM&S7

Cultural and Political Requirements

Cultural Requirements

This requirement does not apply to our project.

Political Requirements

This requirement does not apply to our project.

Requirement # C&P2

This requirement does not apply to our project.

Legal Compliance Requirements

Requirement # L1

Description: System must be FDA compliant

Rationale: We need to be compliant with current laws and regulations.

Fit Criterion: Approval from our legal department

Acceptance Tests: TL1

<u>Requirement # L2</u>

Description: System must be compliant with local privacy laws

Rationale: We need to be compliant with current laws and regulations.

Fit Criterion: Approval from our legal department

Acceptance Tests: TL1

<u>Requirement # L3</u>

Description: System must be compliant with local accounting laws.

Rationale: We need to be compliant with current laws and regulations.

Fit Criterion: Approval from our legal department

Acceptance Tests: TL1

<u>Requirement # L4</u>

Description: System must be compliant with local communications and networking laws.

Rationale: We need to be compliant with current laws and regulations.

Fit Criterion: Approval from our legal department

Acceptance Tests: TL1

Standards Requirements

This requirement does not apply to our project.

Requirements Acceptance Tests

Test# TDT1

Description: Cross-reference all money related info on the website to accommodate for all. Checks user profile data and sees if currency and bought items are accurate

Test Passed: On completion of 1000 tests, have a success rate of 99% or higher.

Test Fail: On completion of 1000 tests, have a success rate of lower than 99%.

Test# TDT2

Description: Access to database. Checks users' play history to see if accurate

Test Passed: On completion of 1000 tests, have a success rate of 99% or higher.

Test Fail: On completion of 1000 tests, have a success rate of lower than 99%.

Test# TDT1

Description: Speed test with a timed deadline. Will be tested 1000 times in one setting to get the most accurate results. Will be tracked by the system itself.

Test Passed: On completion of 1000 tests, have a success rate of 97% or higher.

Test Fail: On completion of 1000 tests, have a success rate of lower than 97%.

Test# TP1

Description: Speed test with a timed deadline. Will be tested 1000 times in one setting to get the most accurate results. Will be tracked by the system itself.

Test Passed: On completion of 1000 tests, have a success rate of 97% or higher.

Test Fail: On completion of 1000 tests, have a success rate of lower than 97%.

Test# TP2

Description: Cross-reference all money related info on the website to accommodate for all.

Test Passed: All currencies are accurate to two decimal spots, if possible.

Test Fail: If there exists a currency that is not accurate

Test# TP3

Description: Use VPNs to check the time in various places around the globe to compare with a third-party universal clock.

Test Passed: Times are accurate to ten seconds, give or take.

Test Fail: Times are not accurate for more than a ten-second leeway.

Test# TP4

Description: Amount of approximate resources for a specific capacity will be calculated and then they will be used up, forcefully. System should be able to run even with said amount of resources used up.

Test Passed: System is still 95% functional with resources used up.

Test Fail: System functionality for said limit is below 95%.

Test# TD1

Description: Ability of System to be able to be run with minimal resources

Test Passed: System running even with 25% or less resources down

Test Failed: System not able to run with 25% or less resources down

Test# TD2

Description: Ability of System to retain user data

Test Passed: No issue causes any loss of user data

Test Failed: Issue causes loss of user data

Test# TD3

Description: Practice runs will be done for the actors that will help do the fixes

Test Passed: All problems can be solved in the given timeframe

Test Failed: Problems cannot be solved and result in user's inability to use product

Test# TD4

Description: Test for system to run despite any potential issues. The problems will be replicated to see if the system is sustainable in said conditions.

Test Passed: System can run with 95% functionality

Test Failed: System cannot run with 95% functionality

Test#TM&S1

Description: Maintenance request with a timed deadline. Request will be submitted to the system. The system will log a on submit time stamp. Once completed the actor will submit an on complete time stamp.

Test Passed: On complete time stamp is within 72 hours of on submit time stamp

Test Fail: On complete time stamp is **not** within 72 hours of on submit time stamp

Test#TM&S2

Description: System down for maintenance. The system is set to maintenance mode and the system administrator logs the time.

Test Passed: System is operational within 5 hours of the logged time.

Test Fail: System is **not** operational after 5 hours of the logged time.

Test#TM&S3

Description: Access to system administration. Authorized and unauthorized users will try to log into the Gameserve system.

Test Passed: Users with system administrator access can log in. Users without access credentials cannot.

Test Fail: At least one user without system administrator credentials logs into Gameserve.

Test#TM&S4

Description: UI and game play tutorials. User watches the tutorials once.

Test Passed: Users is able to play a game, login, logout, access the game lobby and see its profile.

Test Fail: Users is **not** able to play a game, login, logout, access the game lobby and see its profile.

Test#TM&S5

Description: Test browser compatibility. User will enter the game webpage in the following web browsers Google Chrome, Mozilla Firefox, Opera and Apple Safari.

Test Passed: Users is able to play a game, login, logout, access the game lobby, see and set its profile on the browser.

Test Fail: Users is **not** able to play a game, login, logout, access the game lobby, see and set its profile on the browser.

Test#TM&S6

Description: Test mobile platform compatibility. User will open the downloaded mobile application on their mobile device.

Test Passed: Users is able to play a game, login, logout, access the game lobby, see and set its profile on their device.

Test Fail: Users is **not** able to play a game, login, logout, access the game lobby and see and set its profile on their device.

Test#TM&S7

Description: Test mobile platform availability. User will download the application to their mobile device from the mobile app stores.

Test Passed: Users is able to download the app and play a game, login, logout, access the game lobby, see and set its profile on their device.

Test Fail: Users is **not** able to download the app and play a game, login, logout, access the game lobby, see and set its profile on their device.

Test#TM&S8

Description: Test system capacity. System will be operating with our current number of company subscribers.

Test Passed: Systems operates at peak performance (CPU usage <= 88%, Memory usage <=60%, network capacity <=90%)

Test Fail: System does **not** meet all criteria for peak performance.

Test#TM&S9

Description: Test system lifespan.

Test Passed: System is operational 98% of the time for 7 years after launch day.

Test Fail: System is **not** operational 98% of the time for 7 years after launch day.

Test# TS1

Description: Create a non-threatening crash report. Send user notification of crash.

Test Passed: On completion of 1000 tests, have a success rate of 97% or higher.

Test Fail: On completion of 1000 tests, have a success rate of lower than 97%.

Test# TS2

Description: Make a new account and check if it is possible to access source code.

Test Passed: User is not able to see source code.

Test Fail: User is able to see source code.

Test# TS3

Description: Make a new account and see if user is able to see their friends' private information.

Test Passed: User is not able to see any private information.

Test Fail: User is able to see private information.

Test# TS4

Description: Create new account and see if terms of service page pops up. If user declines terms of service, user should not be allowed to proceed.

Test Passed: On completion of 1000 tests, have a success rate of 99% or higher.

Test Fail: On completion of 1000 tests, have a success rate of lower than 99%.

Test# TS5

Description: Users should have a delete account button to delete the account. Check if database contains the deleted user after deletion. If user still appears in database, fail.

Test Passed: On completion of 1000 tests, have a success rate of 99% or higher.

Test Fail: On completion of 1000 tests, have a success rate of lower than 99%.

Test# TS6

Description: Check a players multiplayer activity such as time spent on app, gameplay activity, and ratings. Fail if information can't be found on.

Test Passed: On completion of 1000 tests, have a success rate of 97% or higher.

Test Fail: On completion of 1000 tests, have a success rate of lower than 97%.

Test# TS7

Description: Attempt to breach security, if breach successful, fail.

Test Passed: On completion of 1000 tests, have a success rate of 99% or higher.

Test Fail: On completion of 1000 tests, have a success rate of lower than 99%.

Test# TU&H1

Description: User Changes the avatar

Test Passed: The avatar successfully changes permanently.

Test Fail: The avatar stays as the old one.

Test# TU&H2

Description: User clicks on the "how to" button on main screen.

Test Passed: The game retrieves the how to document and offers the player for a practice game with bots.

Test Fail: The game does not retrieve the document.

Test# TU&H3

Description: User changes the overall language.

Test Passed: The game refreshes and shows the requested language and set that language as the preferred language for future games.

Test Fail: The game stays on English or does not update to the next language.

Test# TU&H4

Description: User goes through a training tutorial.

Test Passed: The account is made successfully and first thing showed is mandatory training.

Test Fail: The training module is not visible to the player.

Test# TU&H5

Description: The game is easy to understand.

Test Passed: Middle schoolers can understand and progress in the game.

Test Fail: It is difficult for kids to understand and will not play.

Test# TU&H6

Description: User utilizes the interactive buttons during gameplay.

Test Passed: The buttons are easy to understand and easy to use.

Test Fail: The buttons are confusing and are clustered so they are difficult to use.

Test# TU&H7

Description: Every user should be welcomed.

Test Passed: Users with mild disabilities are also able to understand the game.

Test Fail: The users with disability do not understand the game.

Test# TU&H8

Description: The “term of use” and “in game harassment policy” documents.

Test Passed: The documents must be signed before a user finish making their account and must follow the policy.

Test Fail: Players do not sign the policy documents and they are able to play the game.

Test#TL1

Description: Legal Test. Product documentation and legal summary is submitted to legal team.

Test Passed: Legal team approves the document

Test Fail: Legal team does not approve the document

Test# TLaF1

Description: Check with Board Games 2.0 INC to ensure that most of the standards are being fulfilled regarding any designs or appearance

Test passed: 95% of the standards are being fulfilled.

Test Failed: less than 95% of the standards are being fulfilled.

Test# TLaF2

Description: The product shall test with a certain group of customers where they will go through the whole process of playing 5 games. Then they will be asked to fill a survey to feel their willingness to play the game again.

Test passed: 87% or more of the customers say they would like to play it again.

Test Failed: less than 87% of the customers say they would like to play it again.

Test# TO&E1

Description: Download game on iOS device, android device, and desktop device.

Test Passed: Able to play game on all 3 platforms .

Test Fail: Unable to play on any of the 3 platforms.

Test# TO&E2

Description: Check storage required per device, app should not take up more than 1.3 GB.

Test Passed: On completion of 1000 tests, have a success rate of 97% or higher.

Test Fail: On completion of 1000 tests, have a success rate of lower than 97%.

Requirements – Test Correspondence Summary

Test	Requirements																												
	P1	P2	P3	P4	P5	P6	P7	O&E3	O&E4	M&S1	M&S2	M&S3	M&S4	M&S5	M&S6	M&S7	M&S8	M&S9	M&S10	M&S11	M&S12	M&S13	L1	L2	L3	L4	P8	D1	D2
TD1																											x		
TD2																												x	
TD3																													x
TD4																													x
TP1	x	x	x																										
TP2				x																									
TP3					x																								
TP4						x	x																			x			
TM&S1										x	x																		
TM&S2												x																	
TM&S3													x																
TM&S4														x	x	x													
TM&S5																x													
TM&S6																	x												
TM&S7								x	x									x											
TM&S8																			x	x									
TM&S9																					x								
TL1																						x	x	x	x				

(CONTINUED)

Requirements – Test Correspondence Summary

	LaF1	LaF2	LaF3	LaF4	LaF5	LaF6		
LaF1	x	x						
LaF2			x	x	x	x		

Table 1 - Requirements - Acceptance Tests Correspondence

Design

Design Goals

SV: Identify the important design goals that are to be optimized in the proposed design.

Content

Design goals are important properties of the system to be optimized, and which may affect the overall design of the system. For example computer games place a higher priority on speed than accuracy, and so the physics engine for a computer game may make some rough approximations and assumptions that allow it to run as fast as possible while sacrificing accuracy, whereas the physics calculations performed by NASA must be much more rigorously correct, even at the expense of speed.

Note an important difference between design goals and requirements: Requirements include specific values that must be met in order for the product to be acceptable to the client, whereas design goals are properties that the designers strive to make "as good as possible", without specific criteria for acceptability. (Note also that the same property may appear in both a requirement and a design goal, so a design goal may be to make the system run as fast as possible, with a requirement that says any speed below a certain specified threshold is unacceptable.)

Your text goes here . . .

Current System Design

*SV: **IF** the proposed new system is to replace an existing system, then the current system should be described here. Otherwise insert a brief statement that there is no pre-existing system.*

Your text goes here . . .

Proposed System Design

This section will make heavy use of class diagrams, and also sequence and deployment diagrams where noted. However don't overlook finite state, activity, communication, or other diagram types as needed for effective communication.

Initial System Analysis and Class Identification

SV: Perform grammatical and similar analyses to identify the most import and obviously needed classes, and to organize them into an initial class structure. An initial class diagram is appropriate, containing few if any internal details.

Your text goes here . . .

Dynamic Modelling of Use-Cases

SV: Insert sequence diagrams of (at least the most important) use-cases, as a means of identifying other needed classes.

Content

Include sequence diagrams of each important use-case here. This is a first step towards identifying preliminary objects. (If the sequence diagram would be too big to fit, then it can either be broken down into pieces or a communication diagram can be used in its place.)

Your text goes here . . .

Proposed System Architecture

SV: Identify the Software Architecture to be applied to this project, such as Client-Server, Repository, MVC, etc., along with justification for the choice.

Your text goes here . . .

Initial Subsystem Decomposition

SV: A slightly more detailed class diagram, showing the classes identified in sections 0, 0, and 0 above, partitioned into subsystems. For each subsystem provide a brief description of the subsystem, including its key responsibilities. There should still be few if any internal details.

Your text goes here . . .

Additional Design Considerations

SV: The sections listed here do not need to be presented in the order given, and may not all be relevant for any particular project. Those that are relevant can help identify additional classes that are needed as a result.

Hardware / Software Mapping

SV: This is particularly important for distributed systems, such as those employing a client-server architecture. Use a deployment diagram to indicate which subsystems are mapped onto which piece(s) of hardware, and what communication subsystems need to be added to the system as a result.

Your text goes here . . .

Persistent Data Management

SV: Document the classes and perhaps subsystems necessary to store persistent data when the system shuts down, and to restore that data when the system starts back up again.

Reiterate key data structures and information as necessary for the understanding of this design phase. Refer the reader back to the data dictionary in section 0 above to avoid undue repetition, while reviewing only the most relevant items here.

Your text goes here . . .

Access Control and Security

SV: Identify the access control and security concerns for this system, and the new classes and/or subsystems that must be added to handle those concerns.

Your text goes here . . .

Global Software Control

SV: Identify the global software control concerns for this system, and the new classes and/or subsystems that must be added to handle those concerns.

Your text goes here . . .

Boundary Conditions

SV: Identify the boundary condition concerns for this system, and the new classes and/or subsystems that must be added to handle those concerns. In particular consider startup, shutdown (normal or abnormal), and the creation and/or maintenance of any configuration files, databases, or similar supporting data files.

Your text goes here . . .

User Interface

SV: Include a preliminary user interface design here, possibly as a rough sketch or other mockup, in order to identify additional classes needed to implement the interface.

The final user interface design will normally be developed by appropriate experts in that area. However it is appropriate to include an initial design here, including possibly a low- or high- fidelity sketch/mockup, in order to identify key classes necessary to implement the user interface, such as forms and dialog windows. It may also go towards addressing usability and/or look-and-feel requirements, and/or identifying other overlooked components.

Your text goes here . . .

Application of Design Patterns

SV: Any design patterns applied as a result of previous sections should have been addressed there, and identified as such at the time. Use this section to document only the additional design patterns that were not previously covered elsewhere. (If any.)

Your text goes here . . .

Final System Design

SV: Include here the final version of the overall system design, incorporating all the subsystems and classes added as a result of additional design considerations. Multiple diagrams may be needed, possibly starting with an overall package diagram showing all the different subsystems and the (important) classes contained within each one. Still not a lot of internal details.

Your text goes here . . .

Object Design

This section documents the internal details of each class, to the extent that they can be designed at this time. Included should be the class interfaces (public method signatures and responsibilities) and constraints. It is probably best to break this section up into subsections corresponding to subsystems as documented above, and/or by (Java) packages if those are designed. It may also be appropriate to address additional design pattern considerations here, but not to the point of being redundant of previous documentation.

Certain methods, such as simple getters, setters, and constructors are not always documented, unless there is something special about them such as in the Singleton or Factory Method design patterns.

Packages

SV: If the design involves assigning classes to packages (.e.g Java packages), then the packages to be created should be documented here.

Your text goes here . . .

Subsystem I

Your text goes here . . .

Subsystem II

Your text goes here . . .

etc.

Your text goes here . . .

Project Issues

Open Issues

SV: Issues that have been raised and do not yet have a conclusion.

Content

A statement of factors that are uncertain and might make significant difference to the product.

Motivation

To bring uncertainty out in the open and provide objective input to risk analysis.

Examples

Our investigation into whether the new version of the processor will be suitable for our application is not yet complete.

The government is planning to change the rules about who is responsible for gritting the motorways, but we do not know what those changes might be.

Considerations

Are there any issues that have come up from the requirements gathering that have not yet been resolved? Have you heard of any changes that might occur in the other organizations or systems on your context diagram? Are there any legislative changes that might affect your system? Are there any rumors about your hardware or software suppliers that might have an impact?

Your text goes here . . .

Off-the-Shelf Solutions

SV: Discussion of products or components currently available that could either be incorporated into the new solution or simply used instead of developing (parts of) the new solution. The distinction between sections 35 a, b, and c is subtle, and not very important.

Your text goes here . . .

Ready-Made Products

SV: Products available for purchase that could be used either as part of a solution or instead of (a part of) a solution.

Content

List of existing products that should be investigated as potential solutions. Reference any surveys that have been done on these products.

Motivation

To give consideration to whether a solution can be bought.

Considerations

Could you buy something that already exists or is about to become available? It may not be possible at this stage to make this determination with a lot of confidence, but any likely products should be listed here.

Also consider whether some products must not be used.

Your text goes here . . .

Reusable Components

SV: Similar to 35a, but for components such as libraries or toolkits instead of fully blown products.

Content

Description of the candidate components, either bought from outside or built by your company, that could be used by this project. List libraries that could be a source of components.

Motivation

Reuse rather than reinvention.

Your text goes here . . .

Products That Can Be Copied

SV: Products that could legally be copied would typically be past projects developed by the same development group, provided there were no restrictions that would prevent their reuse.

Content

List of other similar products or parts of products that you can legally copy or easily modify.

Motivation

Reuse rather than reinvention.

Examples

Another electricity company has built a customer service system. Its hardware is different from ours, but we could buy its specification and cut our analysis effort by approximately 60 percent.

Considerations

While a ready-made solution may not exist, perhaps something, in its essence, is similar enough that you could copy, and possibly modify, it to better effect than starting from scratch. This approach is potentially dangerous because it relies on the base system being of good quality.

This question should always be answered. The act of answering it will force you to look at other existing solutions to similar problems.

Your text goes here . . .

New Problems

SV: The proposed new system certainly has its benefits, but it could also raise new problems. It is a good idea to identify any such potential problems early on, rather than being surprised by them later.

Effects on the Current Environment

SV: Could the new system have any adverse effects on the working environment, e.g. the way people do their jobs?

Content

A description of how the new product will affect the current implementation environment. This section should also cover things that the new product should not do.

Motivation

The intention is to discover early any potential conflicts that might otherwise not be realized until implementation time.

Examples

Any change to the scheduling system will affect the work of the engineers in the divisions and the truck drivers.

Considerations

Is it possible that the new system might damage some existing system? Can people be displaced or otherwise affected by the new system?

These issues require a study of the current environment. A model highlighting the effects of the change is a good way to make this information widely understandable.

Your text goes here . . .

Effects on the Installed Systems

SV: Could the new system have any adverse effects on other hardware or software systems?

Content

Specification of the interfaces between new and existing systems.

Motivation

Very rarely is a new development intended to stand completely alone. Usually the new system must coexist with some older system. This question forces you to look carefully at the existing system, examining it for potential conflicts with the new development.

Your text goes here . . .

Potential User Problems

SV: Could the new system have any adverse effects on the users of the software? Could users possibly have a negative response to the new system?

Content

Details of any adverse reaction that might be suffered by existing users.

Motivation

Sometimes existing users are using a product in such a way that they will suffer ill effects from the new system or feature. Identify any likely adverse user reactions, and determine whether we care about those reactions and what precautions we will take.

Your text goes here . . .

Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

SV: Are there any (physical) limitations in the expected environment that could inhibit the proposed product? (e.g. weather, electrical interference, radiation, lack of reliable power, etc.)

Content

Statement of any potential problems with the new automated technology or new ways of structuring the organization.

Motivation

The intention is to make early discovery of any potential conflicts that might otherwise not be realized until implementation time.

Examples

The planned new server is not powerful enough to cope with our projected growth pattern.

The size and weight of the new product do not fit into the physical environment.

The power capabilities will not satisfy the new product's projected consumption.

Considerations

This requires a study of the intended implementation environment.

Your text goes here . . .

Follow-Up Problems

SV: Basically any other possible problems that could occur.

Content

Identification of situations that we might not be able to cope with.

Motivation

To guard against situations where the product might fail.

Considerations

Will we create a demand for our product that we are not able to service? Will the new system cause us to run afoul of laws that do not currently apply? Will the existing hardware cope?

There are potentially hundreds of unwanted effects. It pays to answer this question very carefully.

Your text goes here . . .

Migration to the New Product

SV: This section only applies when there is an existing system that is being replaced by a new system, particularly when data must be preserved and possibly translated / reformatted. Otherwise just write "Not Applicable" under section 38 and remove sections 38a and 38b.

Requirements for Migration to the New Product

SV: These are a list of requirements relevant to the migration procedures. For example a requirement that the two systems be run in parallel for a time until the client is satisfied with the new system and the users know how to use it.

Content

A list of the conversion activities. Timetable for implementation.

Motivation

To identify conversion tasks as input to the project planning process.

Considerations

Will you use a phased implementation to install the new system? If so, describe which requirements will be implemented by each of the major phases.

What kind of data conversion is necessary? Must special programs be written to transport data from an existing system to the new one? If so, describe the requirements for these programs here.

What kind of manual backup is needed while the new system is installed?

When are each of the major components to be put in place? When are the phases of the implementation to be released?

Is there a need to run the new product in parallel with the existing product?

Will we need additional or different staff?

Is any special effort needed to decommission the old product?

This section is the timetable for implementation of the new system.

Your text goes here . . .

Data That Has to Be Modified or Translated for the New System

SV: This section specifically addresses ***data*** *that must be preserved and/or translated / reformatted during the migration process.*

Content

List of data translation tasks.

Motivation

To discover missing tasks that will affect the size and boundaries of the project.

Fit Criterion

Description of the current technology that holds the data.

Description of the new technology that will hold the data.

Description of the data translation tasks.

Foreseeable problems.

Considerations

Every time you make an addition to your dictionary (see section 5), ask this question: Where is this data currently held, and will the new system affect that implementation?

Your text goes here . . .

Risks

SV: Consideration of the potential risks that could cause the project to fail / underperform.

All projects involve risk—namely, the risk that something will go wrong. Risk is not necessarily a bad thing, as no progress is made without taking some risk. However, there is a difference between unmanaged risk—say, shooting dice at a craps table—and managed risk, where the probabilities are well understood and contingency plans are made. Risk is only a bad thing if the risks are ignored and they become problems. Risk management entails assessing which risks are most likely to apply to the project, deciding a course of action if they become problems, and monitoring projects to give early warnings of risks becoming problems.

This section of your specification should contain a list of the most likely risks and the most serious risks for your project. For each risk, include the probability of that risk becoming a problem. Capers Jones's Assessment and Control of Software Risks (Prentice-Hall, Englewood Cliffs, N.J., 1994) gives comprehensive lists of risks and their probabilities; you can use these lists as a starting point. For example, Jones cites the following risks as being the most serious:

- *Inaccurate metrics*
- *Inadequate measurement*
- *Excessive schedule pressure*
- *Management malpractice*
- *Inaccurate cost estimating*
- *Silver bullet syndrome*
- *Creeping user requirements*
- *Low quality*
- *Low productivity*

• Cancelled projects

Use your knowledge of the requirements as input to discover which risks are most relevant to your project.

It is also useful input to project management if you include the impact on the schedule, or the cost, if the risk does become a problem.

Your text goes here . . .

Costs

SV: An estimate of what it will cost to complete this project. Think not only in terms of dollars, but also time, resources, lost opportunities, etc.

For details on how to estimate requirements effort and costs, refer to Appendix C Function Point Counting: A Simplified Introduction

The other cost of requirements is the amount of money or effort that you have to spend building them into a product. Once the requirements specification is complete, you can use one of the estimating methods to assess the cost, expressing the result as a monetary amount or time to build.

There is no best method to use when estimating. Keep in mind, however, that your estimates should be based on some tangible, countable artifact. If you are using this template, then, as a result of doing the work of requirements specification, you are producing many measurable deliverables. For example:

- *Number of input and output flows on the work context*
- *Number of business events*
- *Number of product use cases*
- *Number of functional requirements*
- *Number of nonfunctional requirements*
- *Number of requirements constraints*
- *Number of function points*

The more detailed the work you do on your requirements, the more accurate your deliverables will be. Your cost estimate is the amount of resources you estimate each type of deliverable will take to produce within your environment. You can create some very early cost estimates based on the work context. At that stage, your knowledge of the work will be general, and you should reflect this vagueness by making the cost estimate a range rather than a single figure.

As you increase your knowledge of the requirements, we suggest you try using function point counting—not because it is an inherently superior method, but because it is so widely accepted. So much is known about function point counting that it is possible to make easy comparisons with other products and other installations' productivity.

It is important that your client be told at this stage what the product is likely to cost. You usually express this amount as the total cost to complete the product, but you may also find it advantageous to point out the cost of the requirements effort, or the costs of individual requirements.

Whatever you do, do not leave the costs in the lap of hysterical optimism. Make sure that this section includes meaningful numbers based on tangible deliverables.

Your text goes here . . .

Waiting Room

SV: This is a place to record ideas or wishes that will not be included in the current release of the product, but which might be worth reconsidering at a later date.

Requirements that will not be part of the next release. These requirements might be included in future releases of the product.

Content

Any type of requirement.

Motivation

To allow requirements to be gathered, even though they cannot be part of the current development. To ensure that good ideas are not lost.

Considerations

The requirements-gathering process often throws up requirements that are beyond the sophistication of, or time allowed for, the current release of the product. This section holds these requirements in waiting. The intention is to avoid stifling the creativity of your users and clients, by using a repository to retain future requirements. You are also managing expectations by making it clear that you take these requirements seriously, although they will not be part of the agreed-upon product.

Many people use the waiting room as a way of planning future versions of the product. Each requirement in the waiting room is tagged with its intended version number. As a requirement progresses closer to implementation, then you can spend more time on it and add details such as the cost and benefit attached to that requirement.

You might also prioritize the contents of your waiting room. "Low-hanging fruit"—requirements that provide a high benefit at a low cost of implementation—are the *highest-ranking candidates for the next release. You would also give a high waiting room rank to requirements for which there is a pent-up demand.*

Your text goes here . . .

Ideas for Solutions

SV: When developing requirements only, it is not the role of the business analyst to dictate the implementation of the solution. However they can pass along any ideas they have here as suggestions to the developers. For CS 440 this report includes system and object design, so this section would make suggestions for implementation and testing that would come after design, such as the use of a particular language, IDE, library, or other tools.

When you gather requirements, you focus on finding out what the real requirements are and try to avoid coming up with solutions. However, when creative people start to think about a problem, they always generate ideas about potential solutions. This section of the template is a place to put those ideas so that you do not forget them and so that you can separate them from the real business requirements.

Content

Any idea for a solution that you think is worth keeping for future consideration. This can take the form of rough notes, sketches, pointers to other documents, pointers to people, pointers to existing products, and so on. The aim is to capture, with the least amount of effort, an idea that you can return to later.

Motivation

To make sure that good ideas are not lost. To help you separate requirements from solutions.

Considerations

While you are gathering requirements, you will inevitably have solution ideas; this section offers a way to capture them. Bear in mind that this section will not necessarily be included in every document that you publish.

Your text goes here . . .

Project Retrospective

SV: At the conclusion of the (CS 440) project, reflect back on what worked well and what didn't, and how the process could be improved in the future.

Content

At the end of every project you should reflect upon what methods were used that worked out well and should be repeated in the future, and also what methods did not work out *well and should be avoided. Any recommendations, suggestions, or ideas for how to do things better in the future should also be documented*

Motivation

To learn from experience, and to continually strive for process improvement.

Considerations

When things don't go well, it is important to distinguish whether the methods themselves were poor, or simply poorly implemented in this particular case, or whether they just weren't right for this particular project / group of engineers.

Your text goes here . . .

Glossary

SV: The glossary is a more complete and inclusive dictionary of defined terms than that found in section I.7.a, the latter of which only covered the most important key terms needed to understand the report.

The glossary defines terms that may not be familiar to all readers. This is especially important if the document is expected to reach a wide and varied audience, such as school children. The glossary may be placed at either the beginning or the end of the document.

Flotsam: *Any part of a ship or its cargo found floating on the water, whether it was deliberately or accidentally lost by its original owners.*

Jetsam: *Any part of a ship or its cargo that is deliberately cast off (jettisoned) by its original owners, generally in order to lighten the ship, whether it floats or sinks.*

Your text goes here . . .

References / Bibliography

This section describes the documents and other sources from which information was gathered. This sample bibliography was generated using the "Insert Citation" and "Bibliography" buttons in the "Citations & Bibliography" section under the "References" tab of MS Word. Creating new citations will not update this list unless you click on it and select "Update Field". You may need to reset the style for this paragraph to "normal" after updating.

[1] Robertson and Robertson, Mastering the Requirements Process.

[2] A. Silberschatz, P. B. Galvin and G. Gagne, Operating System Concepts, Ninth ed., Wiley, 2013.

[3] J. Bell, "Underwater Archaeological Survey Report Template: A Sample Document for Generating Consistent Professional Reports," Underwater Archaeological Society of Chicago, Chicago, 2012.

[4] M. Fowler, UML Distilled, Third Edition, Boston: Pearson Education, 2004.

Index

This section provides an index to the report. The sample below was generated using the "Mark Entry" and "Insert Index" items from the "Index" section on the "References" tab, and can be automatically updated by right clicking on the table below and selecting "Update Field". To remove marked entries from the document, toggle the display of hidden paragraph marks (the paragraph button on the "Home" tab), and remove the tags shown with XE in { curly braces. }