***Ikon Wars Project Report***

**

***Prepared by***

***Nemil Shah, Wishy Parikh, Ruchit Patel, David Qiao***

***for use in* CS 440**

**at the**

**University of Illinois Chicago**

**Table of Contents**

[*REMOVE OR REPLACE ALL TEXT IN RED ITALICS BEFORE SUBMITTING REPORT* 2](#_30j0zll)

[*How to Use This Document* 2](#_3z7bk57)

[List of Figures 9](#_1fob9te)

[List of Tables 10](#_3znysh7)

[I](#_3znysh7) Project Description 11

[1](#_2et92p0) Project Overview 11

[2](#_tyjcwt) The Purpose of the Project 11

[2a](#_3dy6vkm) The User Business or Background of the Project Effort 11

[2b](#_1t3h5sf) Goals of the Project 11

[2c](#_4d34og8) Measurement 12

[3](#_17dp8vu) The Scope of the Work 13

[3a](#_3rdcrjn) The Current Situation 13

[3b](#_26in1rg) The Context of the Work 13

[3c](#_lnxbz9) Work Partitioning 16

[3d](#_35nkun2) Competing Products 18

[4](#_1ksv4uv) The Scope of the Product 18

[4a](#_44sinio) Scenario Diagram(s) 19

[4b](#_2jxsxqh) Product Scenario List 19

[4c](#_z337ya) Individual Product Scenarios 19

[5](#_3j2qqm3) Stakeholders 20

[5a](#_1y810tw) The Client 20

[5b](#_4i7ojhp) The Customer 20

[5c](#_2xcytpi) Hands-On Users of the Product 21

[5d](#_1ci93xb) Maintenance Users and Service Technicians 22

[5e](#_3whwml4) Other Stakeholders 22

[5f](#_2bn6wsx) User Participation 23

[5g](#_qsh70q) Priorities Assigned to Users 24

[6](#_3as4poj) Mandated Constraints 25

[6a](#_1pxezwc) Solution Constraints 25

[6b](#_49x2ik5) Implementation Environment of the Current System 26

[6c](#_2p2csry) Partner or Collaborative Applications 27

[6d](#_3o7alnk) Off-the-Shelf Software 28

[6e](#_23ckvvd) Anticipated Workplace Environment 29

[6f](#_ihv636) Schedule Constraints 30

[6g](#_32hioqz) Budget Constraints 31

[7](#_1hmsyys) Naming Conventions and Definitions 31

[7a](#_41mghml) Definitions of Key Terms 31

[7b](#_2grqrue) UML and Other Notation Used in This Document 32

[7c](#_vx1227) Data Dictionary for Any Included Models 33

[8](#_3fwokq0) Relevant Facts and Assumptions 34

[8a](#_1v1yuxt) Facts 34

[8b](#_4f1mdlm) Assumptions 34

[II](#_2u6wntf) Requirements 36

[9](#_19c6y18) Product Use Cases 36

[9a](#_3tbugp1) Use Case Diagrams 36

[9b](#_37m2jsg) Product Use Case List 38

[9c](#_1mrcu09) Individual Product Use Cases 38

[10](#_46r0co2) Functional Requirements 40

[11](#_2lwamvv) Data Requirements 41

[12](#_111kx3o) Performance Requirements 42

[12a](#_3l18frh) Speed and Latency Requirements 42

[12b](#_206ipza) Precision or Accuracy Requirements 43

[12c](#_4k668n3) Capacity Requirements 44

[13](#_2zbgiuw) Dependability Requirements 45

[13a](#_1egqt2p) Reliability Requirements 45

[13b](#_3ygebqi) Availability Requirements 46

[13c](#_2dlolyb) Robustness or Fault-Tolerance Requirements 47

[13d](#_sqyw64) Safety-Critical Requirements 48

[14](#_3cqmetx) Maintainability and Supportability Requirements 49

[14a](#_1rvwp1q) Maintenance Requirements 49

[14b](#_4bvk7pj) Supportability Requirements 50

[14c](#_2r0uhxc) Adaptability Requirements 51

[14d](#_1664s55) Scalability or Extensibility Requirements 52

[14e](#_3q5sasy) Longevity Requirements 52

[15](#_25b2l0r) Security Requirements 53

[15a](#_kgcv8k) Access Requirements 53

[15b](#_34g0dwd) Integrity Requirements 54

[15c](#_1jlao46) Privacy Requirements 55

[15d](#_43ky6rz) Audit Requirements 56

[15e](#_2iq8gzs) Immunity Requirements 57

[16](#_xvir7l) Usability and Humanity Requirements 57

[16a](#_3hv69ve) Ease of Use Requirements 57

[16b](#_1x0gk37) Personalization and Internationalization Requirements 59

[16c](#_4h042r0) Learning Requirements 60

[16d](#_2w5ecyt) Understandability and Politeness Requirements 61

[16e](#_1baon6m) Accessibility Requirements 62

[16f](#_3vac5uf) User Documentation Requirements 63

[16g](#_2afmg28) Training Requirements 64

[17](#_pkwqa1) Look and Feel Requirements 65

[17a](#_39kk8xu) Appearance Requirements 65

[17b](#_1opuj5n) Style Requirements 66

[18](#_48pi1tg) Operational and Environmental Requirements 67

[18a](#_2nusc19) Expected Physical Environment 67

[18b](#_1302m92) Requirements for Interfacing with Adjacent Systems 68

[18c](#_3mzq4wv) Productization Requirements 69

[18d](#_2250f4o) Release Requirements 70

[19](#_haapch) Cultural and Political Requirements 71

[19a](#_319y80a) Cultural Requirements 71

[19b](#_1gf8i83) Political Requirements 72

[20](#_40ew0vw) Legal Requirements 73

[20a](#_2fk6b3p) Compliance Requirements 73

[20b](#_upglbi) Standards Requirements 74

[21](#_1tuee74) Requirements Acceptance Tests 75

[21a](#_4du1wux) Requirements – Test Correspondence Summary 75

[21b](#_184mhaj) Acceptance Test Descriptions 76

[III](#_3s49zyc) Design 76

[22](#_279ka65) Design Goals 76

[23](#_meukdy) Current System Design 76

[24](#_36ei31r) Proposed System Design 76

[24a](#_1ljsd9k) Initial System Analysis and Class Identification 77

[24b](#_45jfvxd) Dynamic Modelling of Use-Cases 77

[24c](#_zu0gcz) Proposed System Architecture 77

[24d](#_3jtnz0s) Initial Subsystem Decomposition 77

[25](#_1yyy98l) Additional Design Considerations 77

[25a](#_4iylrwe) Hardware / Software Mapping 77

[25b](#_2y3w247) Persistent Data Management 78

[25c](#_1d96cc0) Access Control and Security 78

[25d](#_3x8tuzt) Global Software Control 78

[25e](#_2ce457m) Boundary Conditions 78

[25f](#_rjefff) User Interface 78

[25g](#_3bj1y38) Application of Design Patterns 79

[26](#_1qoc8b1) Final System Design 79

[27](#_4anzqyu) Object Design 79

[27a](#_2pta16n) Packages 79

[27b](#_14ykbeg) Subsystem I 79

[27c](#_3oy7u29) Subsystem II 79

[27d](#_243i4a2) etc. 79

[IV](#_j8sehv) Project Issues 80

[28](#_338fx5o) Open Issues 80

[29](#_1idq7dh) Off-the-Shelf Solutions 80

[29a](#_42ddq1a) Ready-Made Products 81

[29b](#_2hio093) Reusable Components 81

[29c](#_wnyagw) Products That Can Be Copied 81

[30](#_3gnlt4p) New Problems 82

[30a](#_1vsw3ci) Effects on the Current Environment 82

[30b](#_4fsjm0b) Effects on the Installed Systems 83

[30c](#_2uxtw84) Potential User Problems 83

[30d](#_1a346fx) Limitations in the Anticipated Implementation Environment That May Inhibit the New Product 84

[30e](#_3u2rp3q) Follow-Up Problems 84

[31](#_2981zbj) Migration to the New Product 85

[31a](#_odc9jc) Requirements for Migration to the New Product 85

[31b](#_38czs75) Data That Has to Be Modified or Translated for the New System 86

[32](#_1nia2ey) Risks 86

[33](#_47hxl2r) Costs 87

[34](#_2mn7vak) Waiting Room 88

[35](#_11si5id) Ideas for Solutions 89

[36](#_3ls5o66) Project Retrospective 90

[V](#_4kx3h1s) Glossary 90

[VI](#_302dr9l) References / Bibliography 91

[VII](#_1f7o1he) Index 91

### 

### Project Description

## Project Overview

Game named ‘Ikon Wars’ is to be developed two team competes with different avatars aka as characters. The main purpose of this game development is to combine the different qualities of players. For example, the best player is who has education skills like critical and analytical thinking, problem solving skills, interpersonal skills like teamwork, communication, has sarcastic values with how the character is designed etc. The game will enable the player to enhance various skills that he might use in various situations of real life.

The target audience for this game will be for teenagers with age range of 15- 30 years. However, there is no age restriction used. The game is played on online server. Design for this game should contain artistic and creativity for designing characters and making features look more interesting.

Further, we may need to add some problems, puzzles, hints which involves educational values to the game. The title of the game is ‘Ilkon Wars’, but doesn’t have need to have aggressive violence to defeat the player. A player can’t challenge other players with his guns, arms etc, but also with other strategies and powers that can defeat other players without involving violence.

Overall, the project is designed with various features like character build, different levels of the game, real world maps etc. In this game, character designing all will be built before the start of the game. Once the game starts, player gets certain frame time to solve puzzles, riddles and other skills to get the artifacts in their possession. After that, players with similar skill level will be matched in the teams. After that, two teams will be competing.

## The Purpose of the Project

Purpose of this project is to promote the individual growth in all dimensions which includes educational, artistic, and other interpersonal skills.

### The User Business or Background of the Project Effort

There are many online games that have been criticized for the use of violence and brainwashing the minds of teenagers. Because of that many schools in some countries ban that game. Our game should not follow in that category. Whenever any kid or teenager plays the game, no parents or teacher should feel regret about that. The target audience should be very concrete and specific. When all development values are included, audience is wide with less restriction.

Many games are banned with excessive violence which damages the mental health of client to some extend. Our deliver product should enable the player to gain some skills which he/she never had before. Once we reached that, the business will be enabled from all the corners.

Can a student build his aptitude by solving only math problems? Participating in academics is not the only way to increase your talent. Many are struggling in leadership and communication skills. Without solving above problems and visions, the project won’t be started.

Games normally tends to become over additive. Addiction in studying for hours is also harmless. So we are considering certain ways to restrict the user to play the game for only certain hours. Hence, we want the universal acceptance of the game.

### Goals of the Project

Project is carried with vision to sharpen the skills of client in all aspects. The main goal of this game is to combine entertainment values with educational values. Getting artifacts will required the user to solves clues, puzzles and logical reasonings. Further, creating avator will build the creativity side by side. Moreover, during the war, players will require strong teamwork, communication skills and leadership quality. Hence, product is meant for overall personality development of the client.

Special Consideration: This product should also encourage the educational department and social organizations to keep an eye on this.

### Measurement

Several statistics will be made for measuring the success rate for this product. The key measure at the initial stages is to look for how many customers are buying this product in certain time frame. We want to keep track of the graph and see how it is increasing. Next, we want to see that whether the product is accepted universally or not. For example, our age range is from 15-30, but we want to see whether the older people will be able to play or not. Not only this, we also want to check and see whether the game is attracted by other class of people like teachers, elder people, and other critics. We need research team to carry out this process. Other innovative measure we will do is to take short aptitude test for the clients at different stages of the game played. If we somehow able to show that clients are becoming smarter and more intelligent, many doors will be able to market this product in better way.

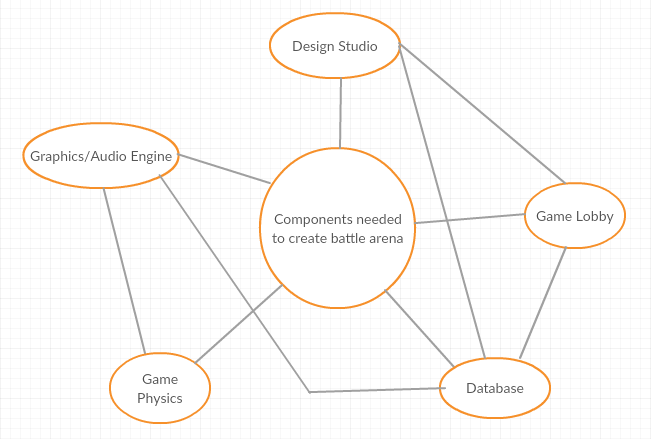
## The Scope of the Work

There are many different maps that users can play on. Each game provides a different experience as there are multiple maps and many different kinds of players can join and play together.

### The Current Situation

This can be published on Steam so; it can be used on Mac and windows. For enthusiasts, this can be costly as it requires playing and winning the games to increase the matchmaking rank to reach on top. These locations can be very large or small. For large maps time limit can be added that will force the avatars to fight with each other. Not only is this, organizing groups of people who can play a virtual game on a real life map near to impossible. In addition, the game will allow users to host an Ikon War server anytime of the day from the comfort of their own homes. Players can save replays of their matches so they can watch it later and learn what they did right or wrong in the match.

### The Context of the Work



### Work Partitioning

|  |  |  |
| --- | --- | --- |
| Event Name | Input and Output | Summary |
| Create Account | User will enter their account info into the database | Allows a player to create and account with his unique ID that will be used in game play |
| Select Map | User will select the map that he wants to play in. | Allows players to download a map from the map database. |
| Find Match | User selects the region that he is in. | Allows players to play in their own region and find players similar to their rank. |
| Start the game | Input: game info from the database.  Output: render information and create game using the game and media engines | When a game is started, information about to the players and the specific scenario being played are sent to the game engines to be rendered and allows the users to play the game. The engines will aid in creating the life-like feel and immersive experience we are aiming for |
| Select Avatar | User will select an Avatar from the database for one game | Allows player to select and apply cosmetics linked to their account. |
| View Profile | Select profile from database | Allows player to check their stats/current progress and watch their old recorded games |
| Watch a game | Server | Allows player to spectator a game that their friends are playing. |

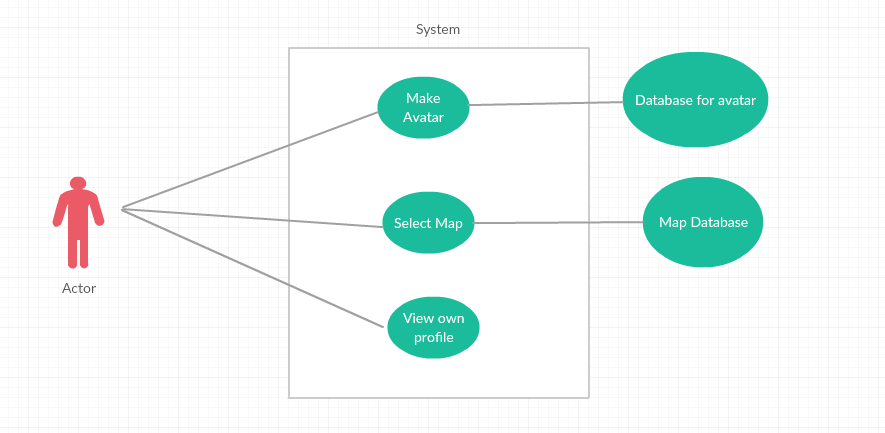
### Competing Products

After doing much research, we found there are other products in MOBA(multiplayer online battle arena) but, we are introducing change in maps and different avatars that are not in the market yet. So it will help players to interact with the real world maps that they live in. Users can create their own virtual world inside their real world maps.

## The Scope of the Product

The main focus of this game is to develop a virtual world inside the real world maps.

### Scenario Diagram(s)



### Product Scenario List

* + - 1. Single Player
      2. Party Game
      3. Bots

### Individual Product Scenarios

Single Player: - In single player game user is matched with other 4 players in the team. Other players will have similar match making rank as the user. Server will find teammates as well as 5 opponents and place them on the map to play.

Party Game: - In party matches user can play with their friends it can be 2, 3, 4 or 5 players. Server will find other players accordingly.

Bots: - In this user will play with the predesigned algorithm which will have 3 levels crusader which will be an easy level. Legend which will be medium level and Immortal will be the hard level.

## Stakeholders

### The Client

The client is the company that are really interest to spread ideas of the map-based game. and motivate how is this interesting while playing. They could start finding target with startup or big company like PS4 or XBox because they have plenty of experience about making games and marketing.

### The Customer

The game customer are the more than average people that love to fight. This game are multiplayer because of that one player connect with other player and make a team while playing game. In addition, Two teams are make challenges and get rank based on who won or lose.

### Hands-On Users of the Product

● User category: Teen age or older.

● User role: Role is play the game

● Subject matter experience: No specific knowledge required to know how to play game, just follow instruction if user is new.

● Technological experience: User need to know how to use computer, but there will also be instruction to guide the user.

● Other user characteristics: The User should enjoy the game with different maps, and different Avatar. Also, some maps are little changing to play. After all, user feels enjoyable and challenging game.

### Maintenance Users and Service Technicians

The people who will work with product maintenance they should know if they find any bug then how to troubleshoot. Also, they know about how to update game.

### Other Stakeholders

Company name: Mixer

Mixer is video game live streaming platform. This company serve large amount of users who enjoy watching other people’s content live. This platform provide good example for streams, who are really interested to playing game. Also, how enjoyable this game and how challenging. In addition, viewers are provide their reviews regarding game.

### User Participation

The user must have computer and some knowledge about how to use computer. There is no additional requirement need accept follow instruction in game.

### Priorities Assigned to Users

● Key users: Simple computer knowledge of use.

● Secondary users: Player will recommend to other players and playing in team is very enjoyable.

● Unimportant users: N/A

## Mandated Constraints

### Solution Constraints

*Description:* The product shall allow to play Ikon game in Single player, Party Games and Bots.

*Rationale:* The client can select avatar and map, this are available with game. There are not other requirements to pay for any extra maps.

*Fit criterion:* Starting menu show display about Single player, Party Game or Bots. User can select anyone. Also, select which map user want to play game.

*Description:* This product work with Window operating system and Mac operating system.

*Rationale:* The client does not have to change their operating system.

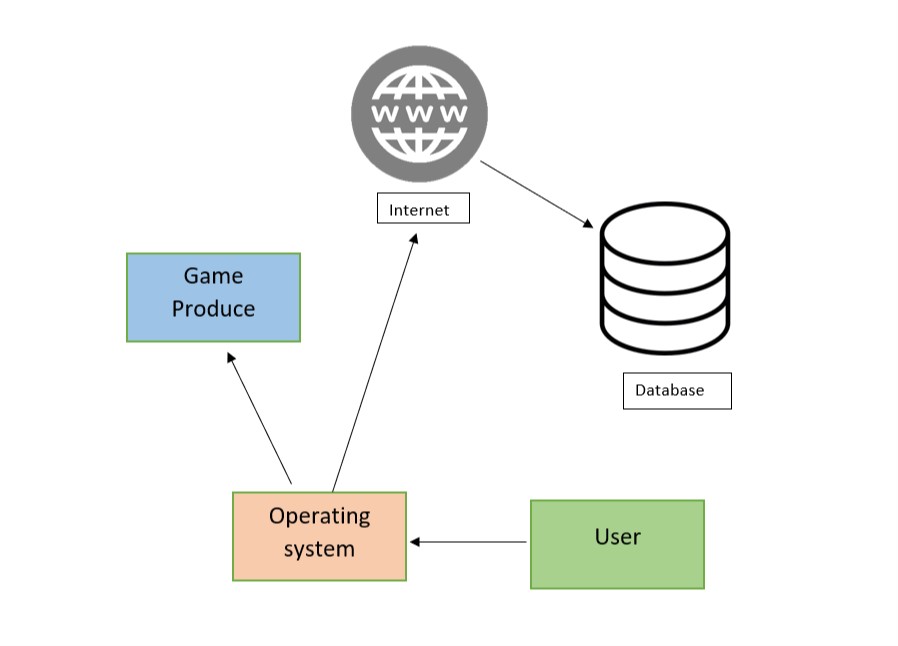
*Fit criterion:* This product connects between Window to Mac or Mac to Window OS. There is not require any particular OS.

*Description:* The product shall allow users to quit in middle of the game, even quit when playing in Party Game, and later resume game. Progress will be saved.

*Rationale:* If Party Game player may not have enough time to play, still they can keep on side their progress and resume later. Between those times, user can play single player game or start Party Game player with different user.

*Fit criterion:* User does not lose any progress with any player.

### implementation Environment of the Current System



### Partner or Collaborative Applications

*Game Engine:* We need connect game engine with Google map to render 3D objects to display map and avatar. We get from another company.

*Game Server:* Product need server. User can save progress and game per player connect with each other and make teams for game, for all this product need server. We can get from another company.

### Off-the-Shelf Software

Content:

Does not need any off-the-shelf software is required for this product.

Motivation:

The product shall handle all requirements such as single player, party game, bots and save progress without depending on any off-the-shelf software.

### Anticipated Workplace Environment

The workplace environment for this product is anywhere. User can access computer/operating system with product install. Internet connection does not require but user want more enjoy with party game than Internet connection is recommended.

### Schedule Constraints

The product requirement will met on its schedule release, as well as any post -release update the product may have.

### Budget Constraints

The budget would depend on project developer team for this product. In Addition rent a database and maintaining after post launch.

## Naming Conventions and Definitions

### Definitions of Key Terms

Player: Human controlled entity, interacts with other players or bots within the map.

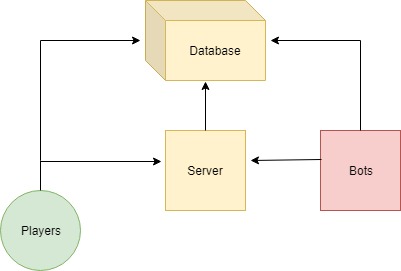
Map: The environment where the match takes place.

Match: A period when entities are allowed to interact with each other. Max players varies by match.

Spectator: Human controlled, allows users to watch matches.

Bot: AI controlled entity, interacts with other players or bots within the map.

### 7b UML and Other Notation Used in This Document



### 7c Data Dictionary for Any Included Models

Database: Holds information of all entities, bots, players, and the match.

Bot Difficulty: Ranges from Crusader to Immortal. (Easy -> Hard)

## Relevant Facts and Assumptions

### 8a Facts

Players and bots alike will aim to bring their team to victory each match.

### 8b Assumptions

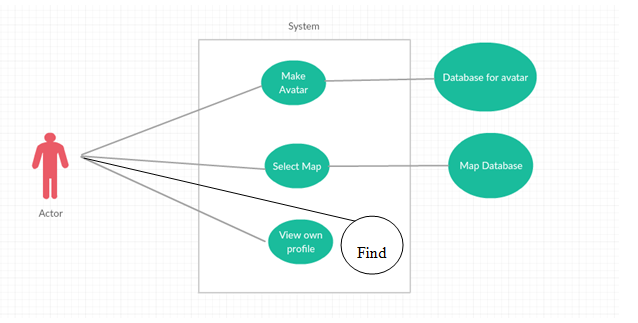
Players will connect to online servers during online ranked matches.

Players will host their own local server if they choose to play solo with bots.

# Requirements

## Product Use Cases

### Use Case Diagrams



### Product Use Case List

* Make Avatar
* Select Map
* View Profile
* Find
  + Normal Match
    - Single Player
    - Party Match
  + Ranked Match
    - Single Player
    - Party Match
  + Bots

### Individual Product Use Cases

|  |
| --- |
| Use Case Name: Make Avatar |
| Actors: User |
| Sequence of Events   1. User can select the Heroes tab from the top of menu 2. User can select what map he wants to play on. |

|  |
| --- |
| Use Case Name: Select Map |
| Actors: User |
| Sequence of Events:   1. User can go in the main Menu to download a map from the database 2. User can select what map he wants to play on. |

|  |
| --- |
| Use Case Name: View User Profile |
| Actors: User |
| Sequence of Events:   1. User can go on the top tabs to view Profile. 2. User can view his stats. 3. User can view his rank. 4. User can view most played heroes and their stats on selected hero. |

|  |
| --- |
| Use Case Name: Find Match |
| Actors: User |
| Sequence of Events   1. User can select Normal Match, Ranked Match, Bots. 2. User has 2 options in normal and ranked matches. 3. In Normal games User is matched with players in region regardless of rank. 4. In ranked games User is matched with the similar ranked players. 5. In Bot games User just plays with the Pre defined lineup. |

## Functional Requirements

**Requirement#:** 1

**Requirement Type** : functional

**Event/Use case#:** 3

**Description:** The system shall allow the user to create an account

**Rationale:** This allows the user to gain access the product’s services and online

capabilities while letting the user personalize their profile

**Fit Criterion:** The option will be available on product start-up

**Requirement#:2**

**Requirement Type** : functional

**Event/Use case#:** 3

**Description:** The system shall allow the user to login with an existing account

**Rationale:** The user needs to access to their saved progress or work in-game

**Fit Criterion:** The option will be available on product start-up

**Requirement#:3**

**Requirement Type** : functional

**Event/Use case#:** 1

**Description:** The System should allow to select Heroes from the database

**Rationale:** The user needs to access database to select available heroes.

**Fit Criterion:** The option will be available on product start-up

**Requirement#:4**

**Requirement Type** : functional

**Event/Use case#:** 2

**Description:** The system shall allow user to select map from the database

**Rationale:** The user needs to access the database to download available maps

**Fit Criterion:** The option will be available on product start-up

**Requirement#:5**

**Requirement Type** : functional

**Event/Use case#:** 4

**Description:** The system shall allow user to find the match in their region

**Rationale:** The user needs to access to the server to find the match

**Fit Criterion:** The option will be available on product start-up

**Requirement#:6**

**Requirement Type** : functional

**Event/Use case#:** 4

**Description:** The system shall allow the user to find match according to rank

**Rationale:** The user needs to get the match according to his rank

**Fit Criterion:** The option will be available on product start-up

## Data Requirements

Two Database are required to fetch information to play the game.

First database is needed to keep the different types of heros. This database is not that space occupied so it will also store user information that will keep track of the stats. Its easy with the heroes because each stat is merged with the heroes.

The first database can also be created to make the AI. Most picked hero, Hero meta can me determined from the same database.

Second database will he for the Maps. This is the one with most of the space because each map is customized according to the virtual game. As the live map has many different places that cannot be utilized for the game. But, the database can store all the maps accordingly.

## Performance Requirements

### Speed and Latency Requirements:

* Game should open quickly within 15 seconds and load all the data that required for the game within next 10 seconds.
* Database for different varieties of avatar must be load in next 5 seconds to allow the user to create his own avator with no lag.
* Database for selecting maps to play the game should be available in 5 seconds.
* After creating avator and selecting maps, player should quick load the game through other submenus within 15 seconds. If the player has to connect online for multiplayer battle, then server should respond in atmost 10 ms ping.
* During the game, there should no lag and game’s status should be updated as efficiently as it can.
* Every match should run with speed, and data and records after the match must be recorded in database.
* User interface for the game should respond back in 1.5 seconds.

### Precision or Accuracy Requirements

* Recorded data after the match and player’s profile must be accurate.
* Database for maps and avators should be precisely available even after updates.
* Database systems to record the played matches, total games bought by the players, total players join in the server should have precise information which must be available to the client.
* Server should be available 24/7 to host the game during online battle.
* Game should be played by the user as precisely it was designed in game engine. For example, actions, movements, storage, etc.

### Capacity Requirements

* System should handle all the database for maps and avators and accurate available for the users.
* Game shall cater for 300 simultaneous users within the period from 9:00 A.M. to 11:00 A.M. Maximum loading at other periods will be 150 simultaneous users.
* Server should hold upto 1 million players atleast through different geographical locations.
* Database should have atleast 10 million records entry preserve for the players. If it is in demand, then it should be double over the course of time.
* User must be given 500mb data to store his avator, maps, recording matches, and various statistics.

.

## Dependability Requirements

### Reliability Requirements:

* Game should easily retrieve data from avator and map database without getting server or unknown load problem.
* Game should not crash during the course of the game. Even during the game, where they are multiple players playing should not cause a bug or error to any of the player during the price poll tournaments.
* During the tournaments, if game crashes, then game must be immediately stopped throwing exceptions to all players saying, ‘Game has crashed and resume as soon it is fixed’. However, it should save the current status of all the players accurately and then resume. A careful consideration must be taken since many players have invested money for the tournaments. Even in the course of big problem, tournament must be cancelled, and each player must be refunded money invested for the tournament.
* Player’s data and their own profile must not be misused or corrupted in any sense.

### Availability Requirements

* For offline players, game is available during 24/7 day and throughout the year.
* For online players, it is available during 24/7 day and throughout the year; however, during the heavy load on server or any kind of maintenance requirement, developer must informed users in advance noticing, ‘Game is under maintenance and will be available after 2 days’.
* Server should response user or give any feedback during the problem caused in money priced tournaments.

### Robustness or Fault-Tolerance Requirements

* User must be given a quick option to message to the developer during the crash of the game.
* During the important tournaments, developer must resolve important issues causing crash of the game. Developer team must be available and prepared to resolve those unexpected issues.
* During the weak connections, player must be informed before entering the tournament prompting, ‘Please make sure you have good connection’.
* When the game is under maintenance during big issues, it should be available locally. Developer should informed the exact dates and times when the game is available online again.

### Safety-Critical Requirements

* Statistics and data of all the players must not be corrupted or sell by any third party.
* Players must be restricted for only certain hours to play the game. For example, player may be restricted to play for eights hour a day max. For small children, parents are given authority to impose more restriction. This avoids the addiction of the player.
* Further, weapons in the game must not be brutal enough. If the weapons are violent and brutal in certain course of game, user must be given restriction according to his/her age. For example, if the user is under 15 years, he will not play those games which have little to no violent weapons. This is to make sure the mental health of small children.
* Game must be developed such that it doesn’t cause over heat to the phone, and if it is then user must be prompted immediately.

## Maintainability and Supportability Requirements

### Maintenance Requirements:

* Game must be updated with new avators and maps during certain course of time.
* Game must be updated with new added features every two months to keep the software trending in market.
* Every six months, game must be put under maintenance to eradicate errors, bugs and flaws and improve the performance and sideby keeping more optimized.
* Developer must ensure that there is no duplicate identity of the user, and user’s identity must not be sell in any course.

### Supportability Requirements

* Game must have pages of manual to show how the game works in detail.
* Game must have other practice sessions and tutorials. For online servers, before the player entered in a big price match tournament, every player may be given chance to test their product with connection and audio.
* 24/7 online staff must be available to support the user for issues and problems.
* For abusive language and spams, support staff should do quick action.

### Adaptability Requirements

* Game is supported in machine with dual core processor with four gb ram, and three gb graphics card.
* User can enhance their grahics of the game according to their own machine capability. For example, machine with high ram and high graph card can have more advance visuals and graphics.
* Game is supported on Windows 10, Mac 12, and Android latest version.

### Scalability or Extensibility Requirements

* During launch, game must support volume of 1 million customers.
* A scalability formula will be design during the finishing of the development process. This formula will ensure how much is game trending and how much volume it should support during the next year. Our statistics team will be managing this.
* Database for maps, avatars and profiles of user should be extensible according to the growth of the customers. Everything will be dynamically managed by the scalability formula.
* Server and system requirements should also support the over heat and load during this rush.

### Longevity Requirements

* Game supports player with multiple age groups, with convenience available for all systems and machines. Game has many online district, state and national based tournament available during certain course of times. Awards and price ceremony will be hold personally by developers to accolade the winners.
* Game should span minimum of seven years as per five million budget spend on it.

## Security Requirements

### Access Requirements

**User and Product owner**

**Description:** User only access their own account.

**Rationale:** User select their own Avatar, Map and Scores.

**Fit Criterion:** Developer staff or owner of product can add Maps, Avatar, and other data

**Acceptance Tests:** User can access personal account and owner or Developer select can change additional features.

### Integrity Requirements

**Accident corruption**

**Description:** By accident or someone tries to hack system product able to protect itself.

**Rationale:** User data store as encryption, so its protected very strong.

**Fit Criterion:** The product shall protect itself from internal abuse.

**Acceptance Tests:** Protect information itself.

### Privacy Requirements

The product shall get permission from user before take information. In addition, product owner make promise to user he will not abuse information.

If information will update or change, first notify to user.

The product shall not share any information to any third party. If in future require to share information, first take permission from user to share details.

### Audit Requirements

**Product retain**

**Description:** Product keep many information.

**Rationale:** Product have server information, network and user personal information.

**Fit Criterion:** Product keep information so, Audit is required.

**Acceptance Tests:** Audit by product owner or other organization by law.

### Immunity Requirements

**Security Update**

**Description:** Product connect to internet.

**Rationale:** On internet transferring data, that warns viruses may come.

**Fit Criterion:** Check online antivirus when downloads data.

**Acceptance Tests:** Keep product updated with security update.

## Usability and Humanity Requirements

### Ease of Use Requirements

**First time user:**

**Description:** First time user follows the instruction for use this product.

**Rationale:** For first user know, what is in game.

**Fit Criterion:** User actual play game and enjoy game.

**Acceptance Tests:** User should not miss guided when use first time product.

### Personalization and Internationalization Requirements

**Language Preference:**

**Description:** This product is available globally.

**Rationale:** In USA default language should be English.

**Fit Criterion:** South korea speaking Korean same as many countries so player connect online in same country.

**Acceptance Tests:** Default language should be based on region.

### Learning Requirements

**Learning:**

**Description:** Learn product for first time.

**Rationale:** First time Make avatar and select maps.

**Fit Criterion:** If user follow the tutorial, then easy to understand.

**Acceptance Tests:** Tutorial is not mandatory but highly recommended.

### Understandability and Politeness Requirements

**Understandability:**

**Description:** User can not see details how product is actual work.

**Rationale:** In the product user easily familiar with used symbol.

**Fit Criterion:** This product available globally, so far symbols are available with description when mouse cursor goes over symbols.

**Acceptance Tests:** User should not be thinking bit for what is symbols is doing.

### Accessibility Requirements

**Affects of color and volume:**

**Description:** Change color and give warning before makes to high volume

**Rationale:** User are colorblind and warning for high level volume.

**Fit Criterion:** If users are colorblind than need to color change and if user make high volume than user get warning of high level of volume.

**Acceptance Tests:** Please follow government guideline for this features.

### User Documentation Requirements

**Manual:**

**Description:** Product come with manual.

**Rationale:** How to start with game and for major direction in game.

**Fit Criterion:** Manual is mandatory with product.

**Acceptance Tests:** Manual also available as e-manual on website.

### Training Requirements

**Training/ Tutorial**

**Description:** In the game there is tutorial.

**Rationale:** First time user doesn’t feel uncomfortable.

**Fit Criterion:** In the game tutorial must available that make product familiar.

**Acceptance Tests:** Tutorial should cover every single part of game.

## Look and Feel Requirements

### Appearance Requirements

**Looks:**

**Description:** Color combination and graphics should attractive.

**Rationale:** If color combination not attractive than user feel old fashion.

**Fit Criterion:** Color combination and graphics feel more enjoyable game.

**Acceptance Tests:** First time see product by customer feel this product will be awesome.

### Style Requirements

**Theme:**

**Description:** Theme play significant for while playing game.

**Rationale:** Theme reflect originality of game.

**Fit Criterion:** Theme change while select Avatar. Also, add music that also change as theme.

**Acceptance Tests:** User not feel every avatar is only look difference, also have different powers as well.

**18 Operational and Environmental Requirements**

**18a Expected Physical Environment**

* Game is expected to be played via desktop computer, or laptop. Internet connection is required for online play.
* Used by all age groups
* Indoor leisurely use

**18b Requirements for Interfacing with Adjacent Systems**

* Internet connection is required for online play.
* Other programs can run in the background while Ikon War is running.
* System requirements are not steep, very basic computer can run this product.

**18c Productization Requirements**

* Product will be distributed via .zip file
* Product will be small enough to fit on a single CD
* Product installation will be very user-friendly

**18d Release Requirements**

* If any bugs / glitches are found, patches will be released as soon as possible

**19 Cultural and Political Requirements**

**19a Cultural Requirements**

* Product will not contain any offensive material in respect to religion or ethnicity

**19b Political Requirements**

* Product will contain no restrictions. to be used on all operating systems.

**20 Legal Requirements**

**20a Compliance Requirements**

* Personal information will not be shared, copied, or distributed, as in compliance with Data Protection Act
* Credit to authors of pictures used will be given

**20b Standards Requirements**

* Product will comply with standard videogame distribution laws

**21 System Design**

**21a Design Goals**

Our product is meant to be used universally by the public. Efficient running times and user friendly installation procedure are to be implemented. Any facts stated in game are to be taken as fictional, and have no relevance or connection to any real world events. Product is meant for leisurely use, to pass time, to have fun.

## 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 24a, 24b, 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 I7c 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. }*

Design 61, 63

Requirements 35, 51, 58

Test 64, 65