

# Requirements Documentation

Gill, Surinder  
1308896

Hu, Joshua  
1311940

Lago, Nick  
1302613

December 8, 2015

## Contents

<b>1</b>	<b>Revision History</b>	<b>6</b>
<b>2</b>	<b>Project Drivers</b>	<b>7</b>
2.1	The Purpose of the Project . . . . .	7
2.1.1	The User Business or Background of the Project Effort . . . . .	7
2.1.2	Goals of the Project . . . . .	7
2.2	The Stakeholders . . . . .	7
2.2.1	The Developers . . . . .	7
2.2.2	The Client . . . . .	8
2.2.3	The Customer . . . . .	8
2.2.4	Other Stakeholders . . . . .	8
2.2.5	The Hands-On-Users of the Product . . . . .	8
2.2.6	Personas . . . . .	8
2.2.7	Priorities Assigned to Users . . . . .	9
2.2.8	User Participation . . . . .	9
2.2.9	Maintenance Users and Service Technicians . . . . .	9
<b>3</b>	<b>Project Constraints</b>	<b>10</b>
3.1	Mandated Constraints . . . . .	10
3.1.1	Solution Constraints . . . . .	10
3.1.2	Implementation Environment of the Current System . . . . .	11
3.1.3	Partner or Collaborative Applications . . . . .	11
3.1.4	Off-the-Shelf Software . . . . .	11
3.1.5	Anticipated Workspace Environment . . . . .	12
3.1.6	Schedule Constraints . . . . .	12
3.1.7	Budget Constraints . . . . .	12
3.1.8	Enterprise Constraints . . . . .	12
3.2	Naming Conventions and Terminology . . . . .	13
3.2.1	Definitions of All Terms, Including Acronyms, Used by Stakeholders Involved in the Project . . . . .	13
3.3	Relevant Facts and Assumptions . . . . .	13
3.3.1	Relevant Facts . . . . .	13
3.3.2	Business Rules . . . . .	14
3.3.3	Assumptions . . . . .	14
<b>4</b>	<b>Functional Requirements</b>	<b>15</b>
4.1	The Scope of the Work . . . . .	15
4.1.1	The Current Situation . . . . .	15
4.1.2	The Context of the Work . . . . .	16

4.1.3	Work Partitioning . . . . .	16
4.1.4	Specifying a Business Use Case (BUC) . . . . .	17
4.2	Business Data Model & Data Dictionary . . . . .	17
4.2.1	Business Data Model . . . . .	17
4.2.2	Data Dictionary . . . . .	18
4.3	The Scope of the Product . . . . .	18
4.3.1	Project Boundary . . . . .	18
4.3.2	Product Use Case Table . . . . .	18
4.3.3	Individual Product Use Cases . . . . .	18
4.4	Functional Requirements . . . . .	18
<b>5</b>	<b>Non-functional Requirements</b>	<b>21</b>
5.1	Look and Feel Requirements . . . . .	21
5.1.1	Appearance Requirements . . . . .	21
5.1.2	Style Requirements . . . . .	21
5.2	Usability and Humanity Requirements . . . . .	21
5.2.1	Ease of Use Requirements . . . . .	21
5.2.2	Personalization and Internationalization Requirements . . . . .	21
5.2.3	Learning Requirements . . . . .	21
5.2.4	Understandability and Politeness Requirements . . . . .	21
5.2.5	Accessibility Requirements . . . . .	21
5.3	Performance Requirements . . . . .	21
5.3.1	Speed and Latency Requirements . . . . .	21
5.3.2	Safety-Critical Requirements . . . . .	22
5.3.3	Precision or Accuracy Requirements . . . . .	22
5.3.4	Reliability and Availability Requirements . . . . .	22
5.3.5	Robustness or Fault-Tolerance Requirements . . . . .	22
5.3.6	Capacity Requirements . . . . .	22
5.3.7	<b>Instability Requirements</b> . . . . .	22
5.3.8	Scalability or Extensibility Requirements . . . . .	22
5.3.9	Longevity Requirements . . . . .	22
5.4	Operational and Environmental Requirements . . . . .	22
5.4.1	Expected Physical Environment . . . . .	22
5.4.2	Requirements for Interfacing with Adjacent Systems . . . . .	23
5.4.3	Productization Requirements . . . . .	23
5.4.4	Release Requirements . . . . .	23
5.5	Maintainability and Support Requirements . . . . .	23
5.5.1	Maintenance Requirements . . . . .	23
5.5.2	Supportability Requirements . . . . .	23
5.5.3	Adaptability Requirements . . . . .	23
5.6	Security Requirements . . . . .	23

5.6.1	Access Requirements . . . . .	23
5.6.2	Integrity Requirements . . . . .	23
5.6.3	Privacy Requirements . . . . .	24
5.6.4	Audit Requirements . . . . .	24
5.6.5	Immunity Requirements . . . . .	24
5.7	Cultural Requirements . . . . .	24
5.8	Legal Requirements . . . . .	24
5.8.1	Compliance Requirements . . . . .	24
5.8.2	Standards Requirements . . . . .	24
<b>6</b>	<b>Project Issues</b>	<b>25</b>
6.1	Open Issues . . . . .	25
6.2	Off-the-shelf Solutions . . . . .	25
6.2.1	Ready-Made Products . . . . .	25
6.2.2	Reusable Components . . . . .	25
6.2.3	Products That Can Be Copied . . . . .	25
6.3	New Problems . . . . .	25
6.3.1	Effects on the Current Environment . . . . .	25
6.3.2	Effects on the Installed Systems . . . . .	25
6.3.3	Potential User Problems . . . . .	26
6.3.4	Limitations in the Anticipated Implementation Environment That May Inhibit the New Product . . . . .	26
6.3.5	Follow-Up Problems . . . . .	26
6.4	Tasks . . . . .	26
6.4.1	Project Planning . . . . .	26
6.4.2	Planning of the Development Phases . . . . .	26
6.5	Migration to the New Product . . . . .	27
6.5.1	Requirements for Migration to the New Product . . . . .	27
6.5.2	Data That Has to Be Modified or Translated for the New System . . . . .	27
6.6	Risks . . . . .	27
6.7	Costs . . . . .	27
6.8	User Documentation and Training . . . . .	27
6.8.1	User Documentation Requirements . . . . .	27
6.8.2	Training Requirements . . . . .	27
6.9	Waiting Room . . . . .	27
6.10	Ideas for Solutions . . . . .	27

## List of Figures

1	Implementation Environment . . . . .	11
---	--------------------------------------	----

2	Activity Diagram . . . . .	15
3	Swimlane Diagram . . . . .	16
4	Business Data Model . . . . .	17

## List of Tables

1	Revision History: Proof of Concept Plan . . . . .	6
2	Definitions . . . . .	13
3	Work Partitioning Part I . . . . .	16
4	Work Partitioning Part II . . . . .	17
5	Data Dictionary . . . . .	18
6	default . . . . .	26

## 1 Revision History

Table 1: Revision History: Proof of Concept Plan

DATE	DEVELOPER	CHANGE	REVISION
October 6, 2015	Gill, Surinder	Initial Draft	0
October 6, 2015	Hu, Joshua	Initial Draft	0
October 6, 2015	Lago, Nick	Initial Draft	0
December 7, 2015	Gill, Surinder	Final Document	1
December 7, 2015	Hu, Joshua	Final Document	1
December 7, 2015	Lago, Nick	Final Document	1

## 2 Project Drivers

### 2.1 The Purpose of the Project

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

This project is to redevelop a popular game Flappy Bird and make it compatible with computers. Once the original developer Dong Nguyen took Flappy Bird off of app stores, the countless users no longer had means to download it. This meant that people would have to find similar games to fill the void that ~~flappy-bird~~ **Flappy Bird** left. With this product we will bring a similar game on a platform that is located in every household. Once completed anyone with a computer will be able to play the addicting and challenging game of ~~flappy-bird~~ **Flappy Bird** once again! Our motivation to make this project is to bring back one of the most popular app games in the interest of entertaining millions! This is not a serious problem but does provide a significant business opportunity for our clients. Flappy Bird at its highest point was making 50 000 dollars a day. By taking this game down there is a serious gap where fans of ~~flappy-bird~~ **Flappy Bird** are looking to find it elsewhere! Whether our client wishes to use this to gain profit off of people playing the game itself, or to gain extra traffic to their websites, it is an opportunity that should not be overlooked! The completed project will be a JavaScript game similar to Flappy Bird to be played on web browsers.

#### 2.1.2 Goals of the Project

Our product will be an entertainment driven experience for users who are interested in games that aren't too complex but still contain challenging aspects. The goals for this project are creating a JavaScript game similar to Flappy Bird, documenting all processes involved in creation and getting a website to host our game. This way the client can have a working demonstration for the product we have designed and what their patrons will see when they access the product. The advantage to making this game in JavaScript is that this will make our game very website friendly. With this we can put the game on the internet for our clients to easily relay to their customers. These are all service goals as they allow the user to access our product easier.

### 2.2 The Stakeholders

#### 2.2.1 The Developers

The developers of our product consist of The A Team. Our role will be to completely redevelop the original application, Flappy Bird, into a new and distinct implementation. Our tasks throughout developing the product will be to use consumer feedback to improve on the original design.

### **2.2.2 The Client**

The client of our product is an external entity who is acting as the commissioner and final reviewer of our product prior to its deployment. They are an entity interested in redeploying the game "Flappy Bird" for the general public.

### **2.2.3 The Customer**

The customer of our project is the general public who will consume the game media. The archetypal customer would be a person with access to the internet and an interface which allows them to interact with the game, ergo an individual near the age of majority in a country with access to our product.

### **2.2.4 Other Stakeholders**

- Members of the public

These people will be our consumer base. They require no extraneous knowledge or background as the instructions for the product are basic and intuitive. In addition they require no involvement for the development of the project. As the project is a redevelopment of a previously existing product, the requirements that this stakeholder has been found through real market response and thus requires no further analysis.

### **2.2.5 The Hands-On-Users of the Product**

- University Students

Consumes media

Master consumer of media

Ability to access completed product

Opportunity to access and use product

### **2.2.6 Personas**

- Name: Abradolf Lincoler

Age: 38

Job: Unemployed

Family: Rick Sanchez, Father

Hobbies: Debating moral philosophy

City: Kirkjubæjarklaustur, Iceland



Favourite Food: Turkey feet

Favourite Music: Post Nu-Grunge Revival

Likes: Birds, Fish, Flappy things, Floppy things

Dislikes Requirements Documentation

Holiday Destination: Hamilton, ON, Canada

Attitude to Technology: Mildly Positive

Attitude to Money: Indifferent

### **2.2.7 Priorities Assigned to Users**

#### **Key Users:**

The Client

The Customer

#### **Secondary Users:**

The General Public

#### **Unimportant Users:**

The General Public

### **2.2.8 User Participation**

#### **The Client:**

No input necessary.

#### **The Customer**

No input necessary.

#### **The General Public:**

No input necessary.

### **2.2.9 Maintenance Users and Service Technicians**

#### **Project Maintenance Team**

These users will maintain the relevance of code in the project and the performance of the website and servers.

## 3 Project Constraints

### 3.1 Mandated Constraints

#### 3.1.1 Solution Constraints

Description: The product shall operate on browsers that allow JavaScript to run (i.e. computers, tablets, phones and other devices) through a website hosting server.

Rationale: The client will not be using outdated technology when accessing our website for our project. The client will be preferably using an up-to-date JavaScript client.

Fit criterion: Our game client will be available to clients with browsers that have JavaScript and if it doesn't, will prompt the user to go to the Java website.

Description: The product shall be played only when connected to the internet to load the game data and upload high scores.

Rationale: The client should be connected to a secure internet connection with an upload and download speed that is appropriate to play the product.

Fit criterion: The product shall be approved to play for users if they have a connection with a good enough internet speed that shall be determined with market research and our game requirements.

Description: The product shall resize to different internet browsers and the different sizes/orientation that browsers use.

Rationale: The client would be able to resize their browser and the product shall resize accordingly until it hits a boundary where it cannot get any smaller or bigger.

Fit criterion: The product shall be able to resize to browsers on all devices and sizes where the game is able to be played. The maximum and minimum sizes will be determined through the game's requirements and sprite sizes.

### 3.1.2 Implementation Environment of the Current System

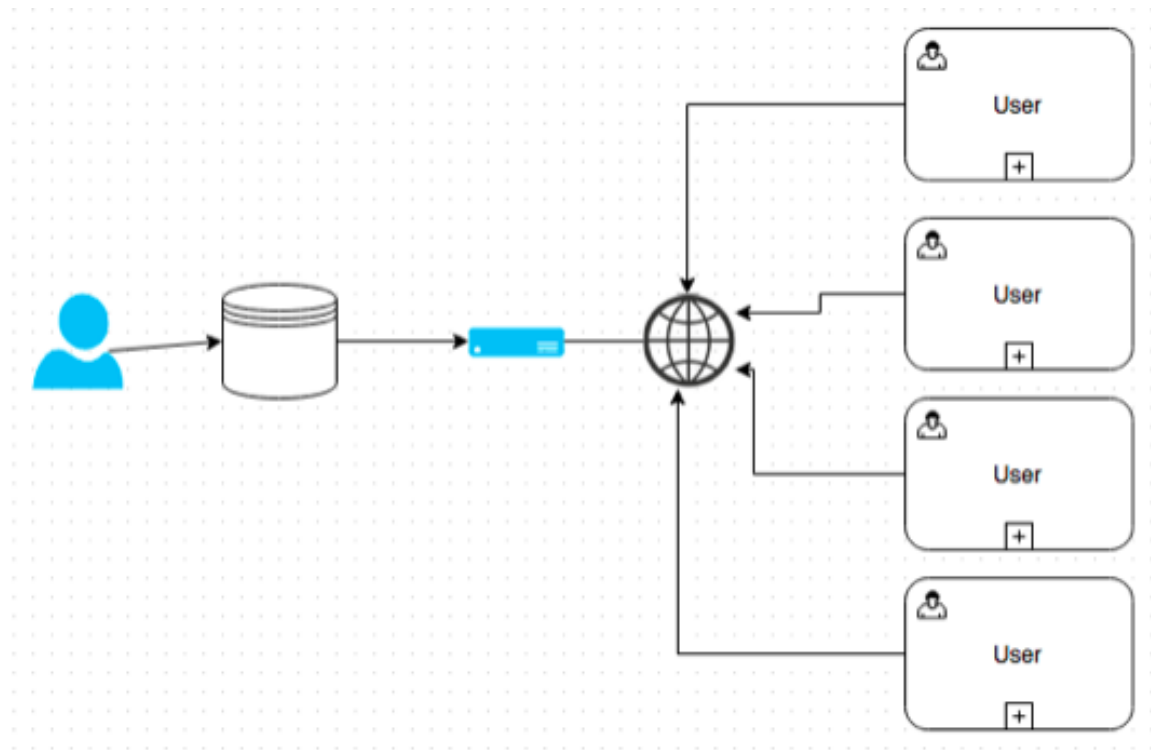


Figure 1: Implementation Environment

### 3.1.3 Partner or Collaborative Applications

The product does not have any direct partner or collaborative applications but does rely on text and picture editors to be created. It also relies on being able to run on a browser supporting JavaScript and if put out on market, a web-hosting server with a domain name.

### 3.1.4 Off-the-Shelf Software

For the product to be executed the following off-the-shelf software is required:

- A web browser (usually comes with the client)
- JavaScript (Available from <https://java.com/en/download/>)
- A client where the web browser can run (i.e. phone, tablet, computer, etc.)

All three of the OTS software can be received for free off the Internet as it is open-source.

### 3.1.5 Anticipated Workspace Environment

The anticipated workplace environment for the product is anywhere. It can be played in someone's house in a permanent location. It can also be played on the go, whether that be on a laptop or mobile device. As long as there is an internet or mobile data connection, the product can be used. It can be inaccessible for clients where they do not have service/internet connection.

### 3.1.6 Schedule Constraints

The schedule constraints are not applicable to our company. The window of opportunity would have been when Flappy Bird was taken down from the mobile market as people would be given a second chance to play the game. However, a deadline has been set to the beginning of December as a chance for our team to have a goal to reach for.

### 3.1.7 Budget Constraints

~~The amount of the budget is not applicable to us~~ A budget is not applicable to this project. We are given all the resources to re-create a project and as a result, ~~are not funding anything~~ a budget is not needed. If there are necessary products to be purchased for the development, it will be an out of the pocket cost to the developers.

### 3.1.8 Enterprise Constraints

This game will be free to play and accessible for any user that has access to a piece of technology that can access a browser. From the browser, they can direct themselves to our project and as long as they have JavaScript, they'll be able to run the project.

### 3.2 Naming Conventions and Terminology

#### 3.2.1 Definitions of All Terms, Including Acronyms, Used by Stakeholders Involved in the Project

Table 2: Definitions

ACRONYM/ABBREVIATION	INTENDED MEANING
JS	JavaScript
HTML	Hypertext Markup Language
PHP	Hypertext Preprocessor
Ex	Example
OS	Operating System
ETC	Et Cetera
bool	A Boolean value
NaN	Not a Number
txt	Text file
git	Git lab
eng	engineer
TAT	The A Team
The A Team	Project Team
WTF	Write to File
DTF	Data Transfer Function
Product	The game that our company is developing
Project	The development of the product
Client	The group that the product is being developed for
Customer	Any individual or group that utilizes the product after completion
The program	The coding that makes up the game created

### 3.3 Relevant Facts and Assumptions

#### 3.3.1 Relevant Facts

Some relevant facts regarding this project are that there were 900 codes of line in the previous implementation of this game. The game has an MIT license and needs to be run through a server for other users to access the project.

### **3.3.2 Business Rules**

One business rule that we've created is that everyone should have the same of amount of work and should try to take initiative at certain areas where they feel stronger.

### **3.3.3 Assumptions**

Some assumptions that developers are making about the project is that all the text and photo editing software will be free and accessible for them, the web-hosting server will be free for launch and the written code will be encapsulated so that it won't be taken and resold for monetary value. Since most electronic devices with Wi-Fi are able to connect to a browser, they'll be able to reach a technological environment in which the product will operate.

## 4 Functional Requirements

### 4.1 The Scope of the Work

#### 4.1.1 The Current Situation

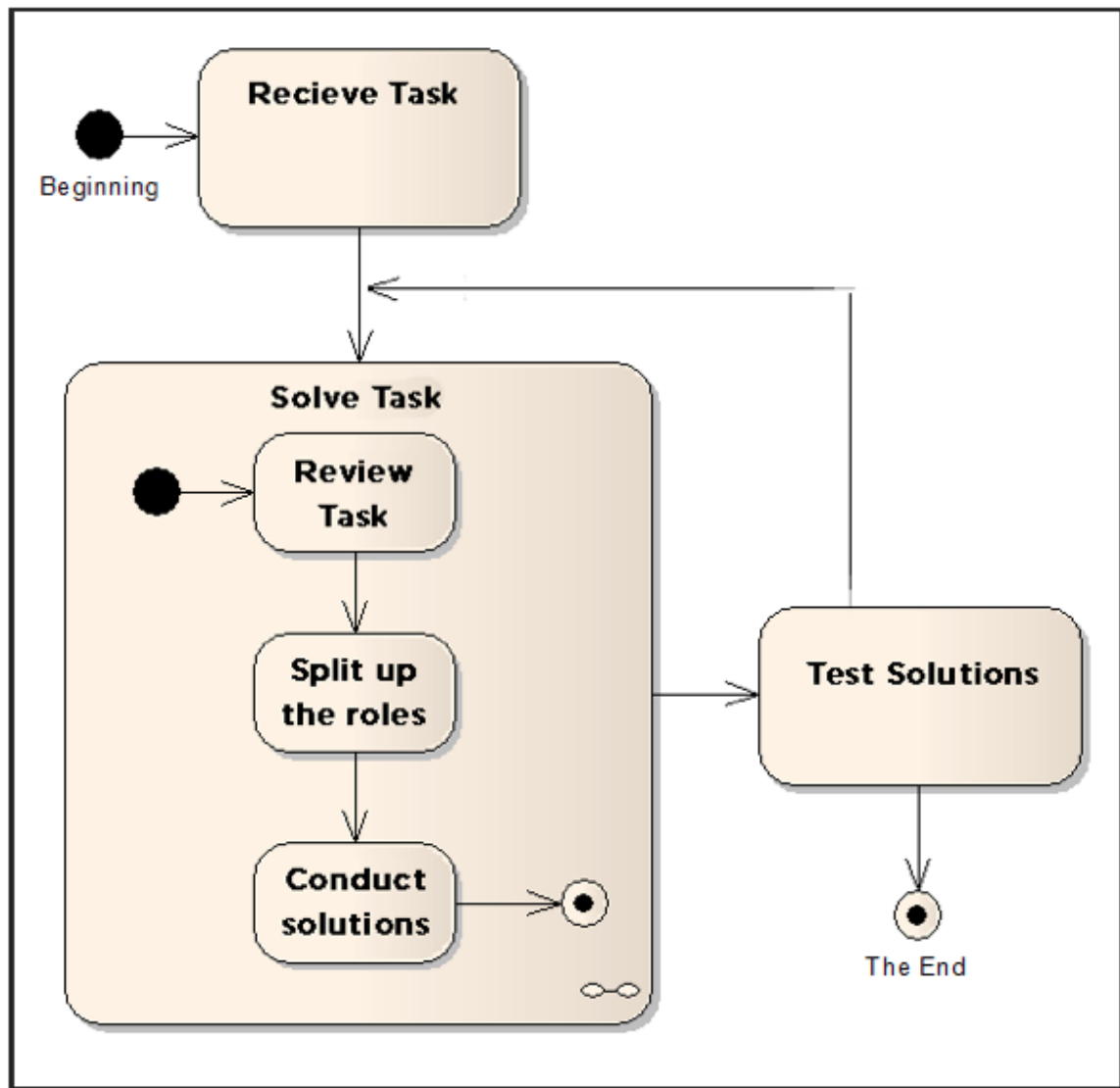


Figure 2: Activity Diagram

#### 4.1.2 The Context of the Work

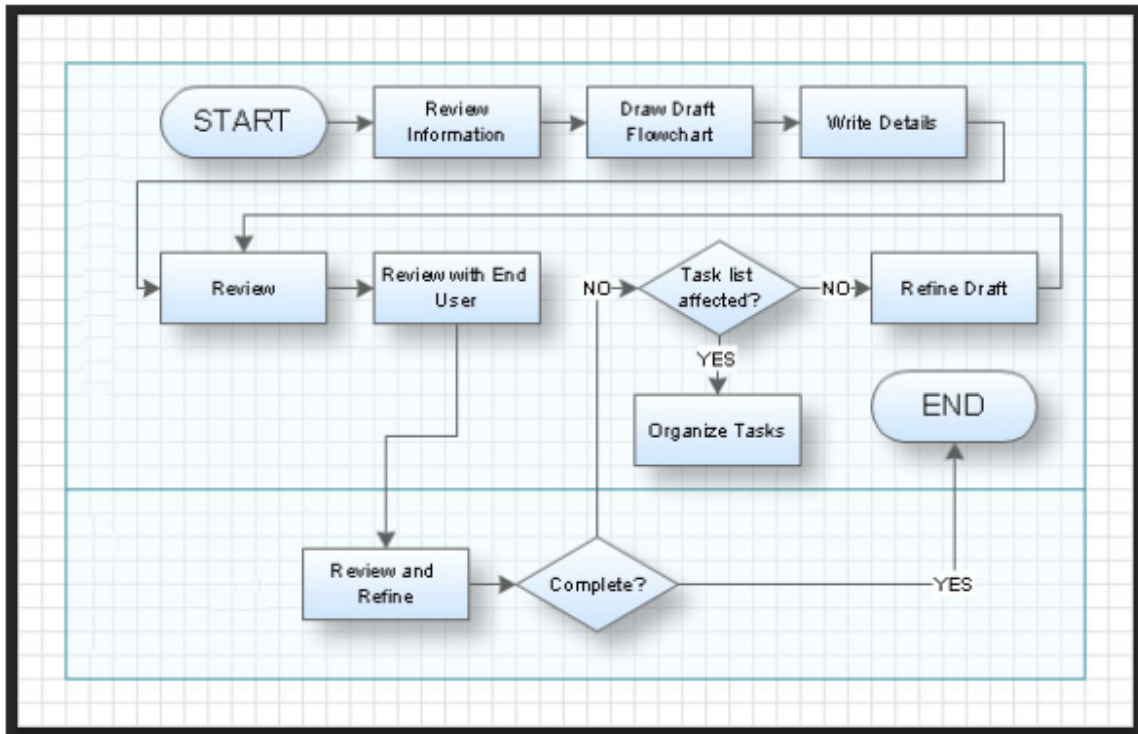


Figure 3: Swimlane Diagram

#### 4.1.3 Work Partitioning

Table 3: Work Partitioning Part I

Event Number	Event Name	Input	Output
1	Floppy Fish Creation	Developer code	Internet Browser
2	Floppy Fish Audio	Microphone	Audio output device
3	Floppy Fish Sprite	Developer graphics and code	Browser
4	Floppy Fish Collisions	Developer code	Internet Browser
5	Floppy Fish Final Edits	Developer code	Internet Browser



Table 4: Work Partitioning Part II

Event Number	Summary of BUC
1	Recreate a browser based app that works on multiple different platforms and saves cookies on browsers
2	Record and edit songs and sounds to be used in the project
3	Create different functions of all the sprites and objects in the project
4	Create collision detection and an end screen
5	Finishing edits to the project

#### 4.1.4 Specifying a Business Use Case (BUC)

Not applicable.

## 4.2 Business Data Model & Data Dictionary

### 4.2.1 Business Data Model

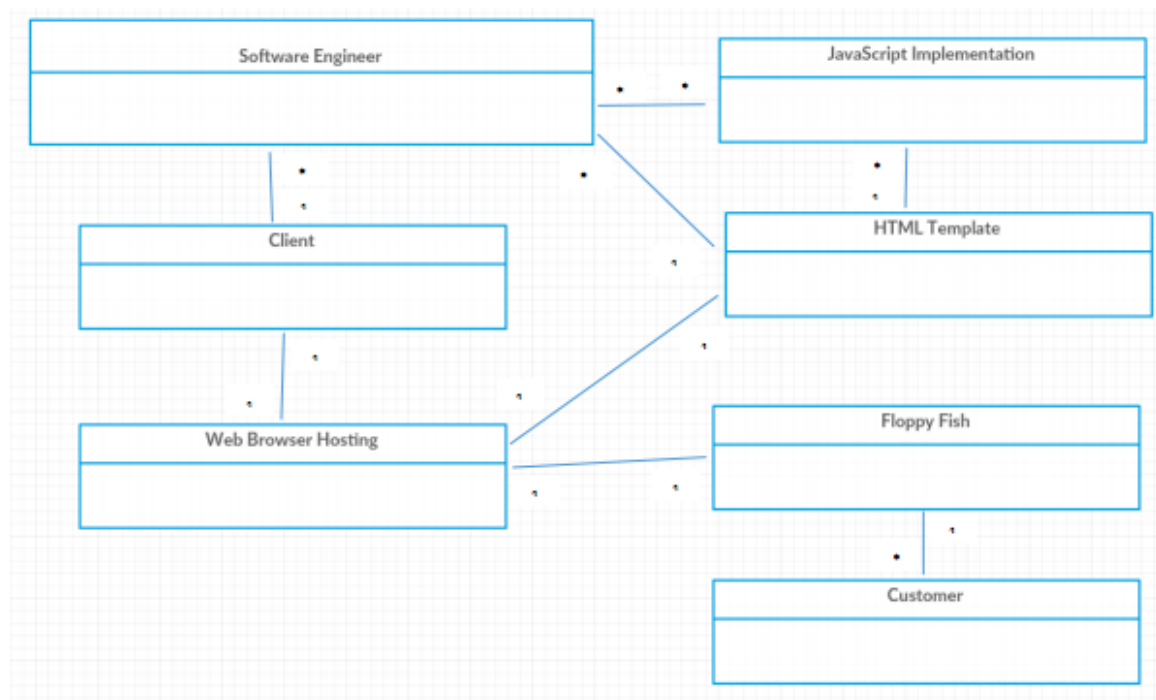


Figure 4: Business Data Model

### 4.2.2 Data Dictionary

Table 5: Data Dictionary

Name	Content	Type
Software Engineer	engineerIdentifier	Class
Client	clientName	Class
Web Browser Hosting	webBrowserIP + webBrowserDomain	Class
Floppy Fish	floppyFishInstance	Class
JavaScript Implementation	isName + jsFunction	Class
HTML Template	templateName	Class
Customer	Name + ID	Class
engineerIdentifier	*Unique Integer Identifier*	Attribute/ Element
ClientName	*String*	Attribute/ Element
webBrowserIP	*IP address*	Attribute/ Element
webBrowserDomain	*Domain Address*	Attribute/ Element
floppyFishInstance	*Created Instance ID*	Attribute/ Element
jsName	*Document Name String*	Attribute/ Element
jsFunction	*Document Description String*	Attribute/ Element
templateName	*Naming String*	Attribute/ Element

## 4.3 The Scope of the Product

### 4.3.1 Project Boundary

### 4.3.2 Product Use Case Table

### 4.3.3 Individual Product Use Cases

## 4.4 Functional Requirements

1. Cookies will be created when the user has reached the GameOver state.

Fit Criterion or Test Case:

Create cookies using the cookie creation function.

2. Cookies will be accessed when the user has terminated his game.

Fit Criterion or Test Case:

Access cookies using the cookie getter function.

3. The executable HTML file will create a new browser window.

Fit Criterion or Test Case:

Is a new browser window created upon the execution of the HTML file?

4. The HTML will be executed by a browser with JavaScript functionality.

Fit Criterion or Test Case:

Attempt to execute the HTML file with multiple major browsers with HTML functionality.

5. The game will have a standby state in which it waits for user input.

Fit Criterion or Test Case:

Execute game and given an arbitrary timeframe if the game does not produces any unexpected action during that timeframe the game does not respond without user input.

6. Upon the reception of user input from the standby state the game will begin.

Fit Criterion or Test Case:

Provide user input to check if the state changes.

7. At the beginning of the game the user will perceive all stats reset to their default state.

Fit Criterion or Test Case:

Return the values of all stats upon the change from the default state.

8. At the beginning of the game the user character will maintain its state until user input is received.

Fit Criterion or Test Case:

Return relative character position at a regular interval prior to and during user input.

9. If there is a collision with the user character and an obstacle object the game will terminate and all stats will be recorded.

Fit Criterion or Test Case:

Given an arbitrary True value of a collision check see if the state changes and return the values of the stats.

10. Upon termination of the game state all stats will be reset to their default state and the standby state will be reinitiated.

Fit Criterion or Test Case:

Return the values of the stats and check the state.

11. If there is a collision with the user character and an objective object the user's score will increment and the objective object's instance will terminate.

Fit Criterion or Test Case:

Return the user's score, check the object's instance.

12. During the game state reception of user input will cause the user character to respond in a constant and uniform manner relative to the user character's instance.

Fit Criterion or Test Case:

Check the response of the user character.

13. Graphic will be loaded and displayed at the correct coordinates.

Fit Criterion or Test Case:

Access source image from the root folder directory or function.

Display coordinates at where the image should be displayed.

14. Individual objects will be created through JavaScript and given colours and/or coordinates.

Fit Criterion or Test Case:

Load inputs of test dimensions, colours and source images and return them.

15. The correct audio sounds will be retrieved and played for their correct duration.

Fit Criterion or Test Case:

Load audio files and play them using the playsound function for their duration.

16. The correct value for a certain calculation is correctly returned.

Fit Criterion or Test Case:

Load inputs for the function and have return the correct value.

## **5 Non-functional Requirements**

### **5.1 Look and Feel Requirements**

#### **5.1.1 Appearance Requirements**

The product shall display the logo of the client company upon starting. Floppy Fish will contain a vivid background of an 'underwater' scene. The product shall generate pipes of a solid colour from a predefined set of colours. Floppy Fish will contain a multicolored bright fish player to catch user's eye while in gameplay. The background shall not overpower any colours of in game objects. The program will be sized to fit properly on a web page.

#### **5.1.2 Style Requirements**

The product will generate pipes in a way that does not make the screen feel cluttered. Flappy Fish will appear to be a bright upbeat game. Flappy Fish shall have a retro style gameplay and fitting. Flappy Fish shall provide the user with a more intense feeling atmosphere rather than laid back.

### **5.2 Usability and Humanity Requirements**

#### **5.2.1 Ease of Use Requirements**

The game shall be able to be played by persons with appendages.

#### **5.2.2 Personalization and Internationalization Requirements**

Not applicable.

#### **5.2.3 Learning Requirements**

The user shall be able to operate an internet browser and a mouse.

#### **5.2.4 Understandability and Politeness Requirements**

The user shall be able to operate an internet browser and a mouse.

#### **5.2.5 Accessibility Requirements**

The game shall be able to be accessed and executed on the majority of user machines.

### **5.3 Performance Requirements**

#### **5.3.1 Speed and Latency Requirements**

The game shall immediately by human perception respond to user input.

**5.3.2 Safety-Critical Requirements**

The game shall not compromise the user data or machine.

**5.3.3 Precision or Accuracy Requirements**

The game shall use integer values.

**5.3.4 Reliability and Availability Requirements**

The game shall be available wherever there is an internet connection at any time when the servers are at or under maximum load.

**5.3.5 Robustness or Fault-Tolerance Requirements**

The game shall be able to operate without consideration of the physical state of the end users.

**5.3.6 Capacity Requirements**

The game shall not exceed the server load.

**5.3.7 Instability Requirements**

The game shall not need to be installed on user's device, but instead downloaded with the webpage and run from the browser.

**5.3.8 Scalability or Extensibility Requirements**

The code will allow for scalability.

**5.3.9 Longevity Requirements**

The product shall be relevant for the lifetime of JavaScript functional browsers.

**5.4 Operational and Environmental Requirements****5.4.1 Expected Physical Environment**

The product is intended to be used anywhere there is a device that has a browser capabilities. The product shall be used in any environment that contains access to the internet.

#### **5.4.2 Requirements for Interfacing with Adjacent Systems**

The product will be run on many different devices as long as the devices can open web browsers properly. The medium that will be carrying our interface is the website hosting our game.

#### **5.4.3 Productization Requirements**

Before the product can be sold or distributed it must be hosted by a website for local use.

#### **5.4.4 Release Requirements**

The product will be revised yearly and updated according to changing demands and needs of the client. The product will undergo maintenance upon realization of any errors in gameplay behaviour.

### **5.5 Maintainability and Support Requirements**

#### **5.5.1 Maintenance Requirements**

Make maintenance of the product minimal.

#### **5.5.2 Supportability Requirements**

Ensure the majority of potential users are capable of running the game on their machines.

#### **5.5.3 Adaptability Requirements**

The game shall be universally accessible by users.

### **5.6 Security Requirements**

#### **5.6.1 Access Requirements**

The entirety of the general public shall be able to access the game provided they have access to the necessary infrastructure.

#### **5.6.2 Integrity Requirements**

The game shall not alter its source code.

The game shall not accept invalid user input.

**5.6.3 Privacy Requirements**

The game shall not divulge data outside the game data.

**5.6.4 Audit Requirements**

Not applicable.

**5.6.5 Immunity Requirements**

Not applicable.

**5.7 Cultural Requirements**

This product shall not be offensive to religious or ethnic groups. This product will be available in English. This product shall be hosted from the North American servers and accommodate ping from all over world.

**5.8 Legal Requirements****5.8.1 Compliance Requirements**

This game shall not compromise any laws.

**5.8.2 Standards Requirements**

This game shall adhere to the MIT Open License.



## 6 Project Issues

### 6.1 Open Issues

~~N/A~~ This game is intended to be used on multiple web browsers and devices which have yet to be tested at this current time. The software and hardware can be differ in the future, resulting in a possibility of unforeseen consequences. This could possibly render our game unplayable or make it interact differently with the user then intended. Additionally, the web server hosting the application could have downtime, too much server traffic or revoke its service which would result in our application being unplayable as it cannot be displayed to our end user. All members of the development team are critical to the development of this project so if any become ill and unable to work, it will cause serious delays in our timeline and maintenance of the project.

### 6.2 Off-the-shelf Solutions

#### 6.2.1 Ready-Made Products

- Servers

#### 6.2.2 Reusable Components

Modularized code components.

#### 6.2.3 Products That Can Be Copied

As an open source project the original implementation can be relied upon as a prototype and source for code.

### 6.3 New Problems

#### 6.3.1 Effects on the Current Environment

The new product will be one of hundreds of millions of different web addresses that host different products. It will not explode the Internet. However, if the server hosting is not able to hold all the users, then it can take down the server and other things being hosted by the server.

#### 6.3.2 Effects on the Installed Systems

This interface stands alone on and doesn't coexist with an older system.

### 6.3.3 Potential User Problems

Some potential user problems as an adverse reaction from playing our project are Carpel-tunnel syndrome and eye soreness/eye strain from extended playing.

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

The server that will host our product will not be powerful enough to hold the desired amount of users with our project growth pattern.

### 6.3.5 Follow-Up Problems

If we one of the developers leave the team, there will be a gap in the information for how the product is implemented. There are also adverse risks if the software becomes too outdated for new upcoming browsers and technology.

## 6.4 Tasks

### 6.4.1 Project Planning

Table 6: default

Task	Completer's Role	Timeline
Model Implementation	Software Engineers	Oct 9th
Model Revision	Client	Oct 10th
JavaScript design	Software Engineers	Oct 30th
HTML implementation	Software Engineers	Nov 3rd
Revision	Client	Nov 5th
Hosting	Software Engineers	Nov 23rd
Maintenance	Software Engineers	Yearly

### 6.4.2 Planning of the Development Phases

Phase 1 Model Implementation: This phase is important as this will be the structure and backbone of our project. This is how we will set up the game and how each module interacts with each other. This should be operational before any coding is started. The product shall be based on this model. Phase 2 Programming Implementation: This phase is where we actually construct the program. Here we will create the game and test it. The program shall be run on JavaScript and HTML. The program will feel as if it is a Flappy Bird inspired game. Phase 3 Hosting and Maintenance: This phase requires us to put the

game onto a server for people to play. At this point for optimal game experience the game shall be maintained at a high standard. This shall be completed and reviewed regularly or upon notice. The program will be continually playable throughout years to come. The program will remain functional as long as the client deems fit.

## **6.5 Migration to the New Product**

### **6.5.1 Requirements for Migration to the New Product**

None.

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

None.

## **6.6 Risks**

None.

## **6.7 Costs**

None as long as the text and image editing software as well as web-hosting and domain name remain free.

## **6.8 User Documentation and Training**

### **6.8.1 User Documentation Requirements**

Not applicable.

### **6.8.2 Training Requirements**

Not applicable.

## **6.9 Waiting Room**

Additional functionality of the game functionality and visual as well as audio effects.

## **6.10 Ideas for Solutions**

Proper heirarchy and documentation of JavaScript code.