Functional Requirements Document
Cranium

Group 13

Zaina Aljallad
Sebastien Beauge
Kevin Gordon
Edwin Ortiz
Abhinav Sharma
Fiona Yip

Version	Description of Change	Author	Date
0.0	Document Created Document formatted	Fiona Yip	02/23/17
0.1	Completed Methodology Completed Functional Requirements	Kevin Gordon	03/03/17
0.2	Completed Background Completed References Completed Assumptions and Restraints	Edwin Ortiz	3/04/17
0.3	Completed Data Flow Diagrams Completed Logical Data Model/Data Dictionary	Kevin Gordon	03/04/17
0.4	Changed document name Completed Other requirements Completed Interface Requirements	Sebastien Beauge	03/05/2017
0.5	Completed Introduction Completed Purpose Completed Scope	Fiona Yip	03/05/17
0.6	Completed Context Completed User Requirements	Zaina Aljallad	03/05/17
0.7	Completed Hardware Interfaces Completed Software Interfaces Completed Communications Interfaces	Abhinav Sharma	03/05/17
0.8	Reviewed Document Formatted/Published Document	Fiona Yip	03/05/17

Table of Contents

1 INTRODUCTION

Cranium is a board game that was created in 1998 by Whit Alexander and Richard Tait. The board game itself is represented like a circuit, with different colors representing different sections, such as, spelling, acting, drawing, or knowing facts. This is what makes Cranium "a game for your whole brain."

1.1 Purpose

This functional requirements document is to serve as a guide in creating a digital version of the game Cranium. This will allow the game to be more accessible for those who want to play the game anytime on any portable device (i.e. smartphone, tablet, laptop, etc.)

1.2 Scope

The scope is to create a functional digital version of the game Cranium. Like the board game, the digital version must also have 4 or more players. Because it is going to be digital, the game must be able to have a "game room" for groups of known players, the ability to find random players, and to have a player versus computers option. The game's interface should be simple and easy to use when it comes to selecting players and an interface for the game itself should be similar to the board.

1.3 Background

The organization group include members all participating in Human and Technology Interaction (CAP 4014) during the Spring semester 2017. The group of people intend to work as a team to accomplish a well informed and established functional requirements document. Portions of the document have been divided equally among each member of the group and is responsible to uphold their share while assisting other members if necessary.

1.4 References

- Meetings Summaries: Discussions via GroupMe to assign responsibilities in creating the requirement document and help answer questions for everyone.
- https://en.wikipedia.org/wiki/Cranium_(board_game)
- https://webcourses.ucf.edu/courses/1249196/files/folder/Lectures?preview=60924464

1.5 Assumptions and Constraints

- A smart phone capable of running the application
- Network Connection to download the game
- Allocated memory to install the game

- Patent to use Cranium License

1.5.1 Assumptions

We have to assume that our target audience are all capable of downloading the application onto their smartphone in addition to having the required memory space available to install it. Also, we must assume that the device is capable of running the program after installation without problems. This could include having enough RAM and is up to date on the latest OS version.

1.5.2 Constraints

Our biggest constraint about our application would be the purchasing of the license to make a game that follows the Cranium patent laws. If this license is not bought before the finish product of this game is released the game can never be distributed to any customers under the fact that the developers could be sued for copyright infringement. Additionally, the application would be available, for the time being, only on smartphones and no browser equivalent.

1.6 Document Overview

The document is divided into four different sections with headings and subheadings. The headings give a brief overview of what is in the section and the sub-heading gives more detail. The sections are organized as follows:
Section 1: The introduction section that gives more general information about the whole project like scope, background and constraints.
Section 2: The methodology selected for the project. The agile methodology was chosen because of the breakdown ability.
Section 3: Looks more into the information needed to create a project like this. It describes how the variables react and the basic functional requirements.
Section 4: Looks at the requirements and other components that was not located in section 3. Things like how the hardware and software for the project interact are in this section.

2 METHODOLOGY

Agile Methodology will be used because of the ability to break down the project into separate smaller tasks that can be done separately from one another.

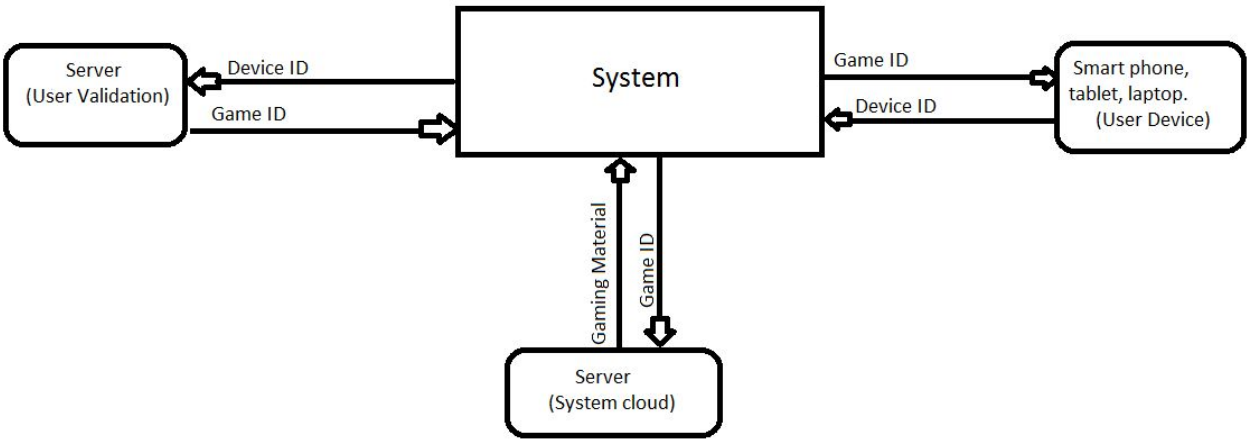
3 FUNCTIONAL REQUIREMENTS

3.1 Context

The User's device will receive the game ID for verification, then the device will send its device ID back to the system. The System cloud will receive the Game ID and send the material in the

game back to the system. After this, the system will communicate to the server which validates the user and sends the device ID which it returns the game ID.

Exhibit 1 - Context Diagram



3.2 User Requirements

Functional Requirements

- Appearance must be consistent. For example: colors should match and be the same throughout the game be familiar to users.
- Text and font shouldn't be obnoxious but noticeable.
- Options should be easy to find and should have many customizable features for users to design the game how they prefer.
- Main game should be as similar to board game as possible to keep the consistency of the game.
- Instructions and tutorials should be easily accessed from any screen and easy to follow for new players.
- If problems occur through server, there will be options to send problems to main server to be able to fix quicker.
- U - User will need smart device and account to play.
- U - User will need at least 4 people to play a game or connect to the internet to play with other users who have the game downloaded.
- A secure network will keep the game from being tampered with.

3.3 Data Flow Diagrams

Device: The device acts as a go between the system and the user. The user initiates the system through the device and starts the application.

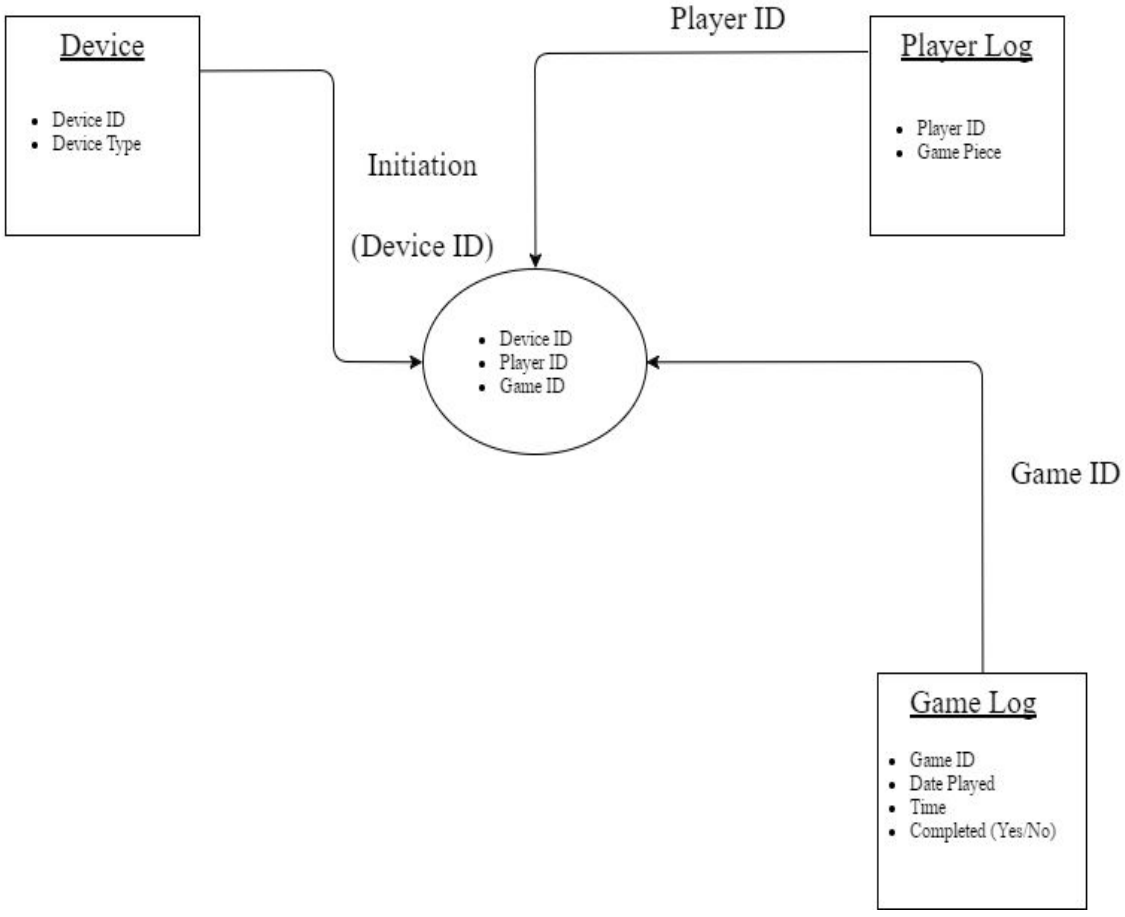
System: Main component of the application and enable other entities to communicate with each other. Receives player and game IDs.

Player Log: Contains player information used for gameplay such as player ID, used for that specific game.

Game Log: A history containing information for all games played, such as date played, time played for, and whether game is completed or not.

3.4 Logical Data Model/Data Dictionary

Logical Data model below shows how the entities and the system relate to one another



3.5

Functional Requirements

Section/Requirement ID	Requirement Detail

FR 1.0.0	System should allow user to define parameters before beginning the game
FR 1.1.0	System should allow between 2 – 4 players, who would serve as team captains
FR 1.2.0	System should create up to four game pieces that players can chose from
FR 2.0.0.0	System should simulate game of Cranium
FR 2.1.0.0	Order of turns shall be determined by order of birthdate from date game is played. Whoever team member has a birthday coming soonest shall go first. Out of the three remaining teams, again whichever team member birthday comes soonest shall be second and so forth.
FR 2.1.1.0	If birthday already occurred in the calendar year, birthday of the next calendar year shall be used
FR 2.2.0.0	Players should take turns in order that was determined.
FR 2.2.1.0	All pieces start in the same, middle purple space
FR 2.2.2.0	A turn starts with a single die roll
FR 2.2.2.1	A die should consist of six sides with colored sides corresponding the following: 2 purple, red, blue, green, yellow
FR 2.2.2.2	System should input the result from die roll
FR 2.2.2.3	System should place game piece to nearest colored space corresponding to result of die roll or the nearest purple space (whichever is closer)
FR 2.2.3.0	System should use space game piece is located on to determine player action
FR 2.2.3.1	Space game piece is located on should be one of five colors: red, blue, green, yellow, and purple
FR 3.0.0	System shall use various functional requirements based on what color space player's game piece is on

<table>
<tr><td>FR 3.1.0</td><td>System shall use these functional requirements related to Red Card when player lands on red colored space on game board</td></tr>
<tr><td>FR 3.1.1</td><td>System shall select randomly from “stack” of red cards and perform associated action</td></tr>
<tr><td>FR 3.1.2</td><td><table>
<tr><th>Red Card (Data Head)</th><th>Requirement Definition</th></tr>
<tr><td>Factoid</td><td>System should present question, and team whose piece landed on red space must answer outright, without hints or choices</td></tr>
<tr><td>Selectaquest</td><td>System should present a question and four possible answers, and team whose piece landed on red space must choose from one of the possible choices. Only one can be correct</td></tr>
<tr><td>Polygraph</td><td>System should present a statement, and team whose piece landed on red space must determine if statement is true or false</td></tr>
</table></td></tr>
<tr><td>FR 3.2.0</td><td>System shall use these functional requirements related to Blue Card when player lands on blue colored space on game board</td></tr>
<tr><td>FR 3.2.1</td><td>System shall select randomly from “stack” of blue cards and perform associated action</td></tr>
<tr><td>FR 3.2.2</td><td><table>
<tr><th>Blue Card (Creative Cat)</th><th>Requirement Definition</th></tr>
<tr><td>Cloodle</td><td>System should give one team member a word or phrase, and</td></tr>
</table></td></tr>
</table>

<table>
<tr><td></td><td>
<table>
<tr><td></td><td>team member must draw on paper what word or phrase is and teammates must guess word or phrase</td></tr>
<tr><td>Sensosketch</td><td>Similar to Cloodle, but drawer has to do task eyes closed</td></tr>
</table>
</td></tr>
<tr><td>FR 3.3.0</td><td>System shall use these functional requirements related to Green Card when player lands on green colored space on game board</td></tr>
<tr><td>FR 3.3.1</td><td>System shall select randomly from “stack” of green cards and perform associated action</td></tr>
<tr><td>FR 3.3.2</td><td>
<table>
<tr><th>Green Card (Star Performer)</th><th>Requirement Definition</th></tr>
<tr><td>Charades</td><td>System should give one team member word or phrase, and team member must act out word or phrase and teammates must guess word or phrase</td></tr>
<tr><td>Impressions</td><td>System should give one team member name of celebrity or fictional person, and team member must either act or speak like them. Teammates must guess who celebrity or famous person is</td></tr>
<tr><td>Humdingers</td><td>System should give one team member name of a well-known song and team member must hum or whistle the tune. Teammates must guess name of song</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td></td></tr>
<tr><td>FR 3.4.0</td><td>System shall use these functional requirements related to Yellow Card when player lands on yellow colored space on game board</td></tr>
<tr><td>FR 3.4.1</td><td>System shall select randomly from “stack” of yellow cards and perform associated action</td></tr>
<tr><td>FR 3.4.2</td><td>
<table>
<tr><th>Yellow Card (Word Worm)</th><th>Requirement Definition</th></tr>
<tr><td>Spelling Challenge</td><td>System gives word to one team member and they must spell it correctly in one try</td></tr>
<tr><td>Gnilleps Challenge</td><td>System gives word to one team member and they must spell it backwards (from last to first letter) in two tries</td></tr>
<tr><td>Blankout</td><td>System gives team a word with blanks with only some letters filled in and team must guess what word is</td></tr>
<tr><td>Lexicon</td><td>System gives team a definition and four choices as to what the matching word is. Team must guess with word matches correctly with the definition. Use rarely used words.</td></tr>
<tr><td>Zelpuz</td><td>System gives team a hint and an anagram of the word and team must rearrange anagram to the correct word</td></tr>
</table>
</td></tr>
</table>

FR 3.5.0	If game piece lands on purple colored space, including at the beginning of the game, team is allowed to choose which of the aforementioned four cards they would like to try
FR 4.0.0	Each challenge has a 10 second time. If a team does not complete the challenge within those 10 seconds, it counts as an unsuccessful try
FR 5.0.0	If team successfully completes their challenge, they can roll the die and move accordingly and wait until their next turn to do next challenge. If not, they stay where their game piece is currently place and end their turn
FR 6.0.0	A space named “Club Cranium” should be included on the game board. If a team lands on this space, they must try one of each of the four challenges, as well as one choice from the team whose turn is coming up. If successful in all challenges, team wins game. If team is unsuccessful in one of the challenges, their turn ends and they must wait until their next turn to try again.

4 OTHER REQUIREMENTS

Multiple players on the same system: The device should be able to support at least four players on the same device.

4.1 Interface Requirements

Upon initial launch, a general menu must be given. Multiple options like if online or same device play must be given, settings and account settings. For account settings, user must enter several parameters.

For same device play, the system needs to know how many players there are, how many teams and even distribution of players. The system needs to notify the user if the teams are unbalanced and can't start the game unless the constraints of the game are met. For online play, it should follow the same setup as same device play. For example, the board should still be visible to everyone, everyone gets a turn and the boars should be visible as well.

After initial teams are formed, user data needs to be entered. User names, date of birth and user icons should also be selected. After all the initial information was entered, a board will be

created based on all the information and based on the functions requirement. Between turns, the system should have a notification to let the user know who's turn is next and show a layout of the board for the user to see. Gameplay rules should follow the functionals requirement to make the game as accurate as possible.

4.1.1 Hardware Interfaces

Cranium will support modern smartphone devices running Android and iOS operating systems. The hardware should at least be capable of running the game without slowdown and runtime issues. For Apple products, devices are recommended to have at minimum an A6 processor with a 1.3GHz dual-core ARM CPU and 1GB of LPDDR2 RAM. This includes every phone past the iPhone 5 made by the manufacturer as well as all renditions of the iPad past the iPad 4. For the Android side of things, systems should have at least or equal to a Snapdragon 801 processor with a Quad-Core 2.5GHz Krait CPU. An example of this is the Samsung Galaxy S5. An Adreno 330 GPU is recommended but not required and although lower versions of hardware may suffice as long as Android Lollipop is run, those versions are also not recommended. Since the game will also support online play, industry-leading devices are suggested to be able to decrease latency between turns due to hardware lag and issues.

4.1.2 Software Interfaces

Due to modern app development, software interfaces should make the game fairly easy to develop. The user will need to register and validate with a cloud server. The server will support the network aspects of the game which includes communication, turn timers, and maintaining game information across all devices. Cranium should be able to be played completely offline on a single user's device. This means that all files and game logic are stored locally on the device. For development, platforms such as Xamarin or Visual Studio should be used to ensure cross-platform support across iOS and Android devices. For online devices, the cloud server dedicated for communication of online games should be able to support a large amount of games at once due to the light load of server-side information needed.

4.1.3 Communications Interfaces

Players should be able to play locally as well as over the internet, on Wi-Fi or mobile data. The network connection should be strong enough to ensure constant communication between the host and the server to allow every user to have a positive experience while playing. For local gameplay, network communications should be allowed to be disabled while playing.

4.2 Data Conversion Requirements

Since this is all newer operating systems, legacy data conversion is not necessary.

4.3 Hardware/Software Requirements

For this project, a smart phone with a stable Internet connection is needed to download the application. The supported platforms will be based on operating systems. Androids should be at least Lollipop or above and iPhones/Apple devices should be at least iOS 8 do download and run the application.