



## PARKinetics

Group Name: TANKER

Group Number 6

<https://ksackvil.github.io/CMPT-275/>

Armaan Bandali

Rachel Djauhari

Negar Hariri

Yuxiang Huang

Takunda Mwinjilo

Kai Sackvile-Hii

## Table of Contents

Revision History .....	2
Software Tools for Automatic Unit Testing .....	3
Internal Deadlines .....	3
Version 1 .....	3
Version 2 .....	3
Version 3 .....	4
User Acceptance Testing .....	4
Integration Testing .....	5
Testing Strategy .....	5
Test Cases .....	5
Software Tools for Measuring Complexity .....	5
Quality Assurance .....	5
References .....	6

## Table of Figures

Table 1: Detailed record of the revision history for this document .....	2
Table 2: Version 1 Internal Deadlines .....	3
Table 3: Version 2 Internal Deadlines .....	3
Table 4: Version 3 Internal Deadlines .....	4
Table 5: Integration Testing (Test Cases) .....	5

## Revision History

*Table 1: Detailed record of the revision history for this document*

Revision	Status	Publication/Revision Date	Author
1.0	First Draft	October 10, 2019	Rachel Djauhari
1.1	Added sections from google doc draft	October 14, 2019	Rachel Djauhari
1.2	Added Internal Deadlines	October 15, 2019	Negar Hariri
2.0	Edited deadlines, inserted captions, and added last 2 sections.	October 15, 2019	Rachel Djauhari
2.1	Added Milestones for Internal Deadlines	October 15, 2019	Negar Hariri

## Software Tools for Automatic Unit Testing

There are various popular iOS test automation frameworks that are available for free such as Appium, XCTest/ XCUITest, Detox, Calabash, EarlGrey [1]. XCTest/XCUITest is a good option for our purposes because it does not require any extra setup or time consumption. Other options are advantageous for cross-platform app automation testing, but they need extra setup and learning costs. OCUit is another test framework that is integrated in XCode, but XCTest and XCUI is preferable because it can be utilized for unit testing and UI automation testing. With regards to the backend server aspect of automation testing, we will be using the Node.js test framework lab considering the experience of the backend developer.

Since the automation test framework is integrated in XCode and all the test codes will be committed to Github under the same project directory, test cases for the iOS app will be written under the same project of the app. Backend automation test cases will be committed to a different project if needed. All other manual execution of test cases will be recorded in another document which will also be committed to Github and test results will also be automatically generated.

## Internal Deadlines

### Version 1

Table 2: Version 1 Internal Deadlines

<i>Tasks</i>	<i>End Date</i>	<i>Milestone</i>
User-Interface	October 28, 2019	- UI test
Firebase (Database)	October 28, 2019	- Perform Server and Database Test
“Finger Twister” Game	October 30, 2019	- Create test cases for this game. - Perform auto and manual tests for this game
Testing Version I	October 31, 2019	- Create Test cases for version I - System test - Perform Unit testing on features.

### Version 2

Table 3: Version 2 Internal Deadlines

<i>Tasks</i>	<i>End Date</i>	<i>Milestone</i>
Preliminary Testin “Shadow DDR” (AR Game)	November 11, 2019	- Perform auto and manual tests for this game
Preliminary Testin “Adventure Story” (Speech Game)	November 11, 2019	- Perform auto and manual tests for this game

Testing Version II	November 15, 2019	<ul style="list-style-type: none"><li>- Create Test cases for version II.</li><li>- Perform Unit test for added features.</li><li>- Perform Unit test for fixed bugs from previous test version.</li><li>- Perform system test</li></ul>
--------------------	-------------------	--

## Version 3

Table 4: Version 3 Internal Deadlines

Tasks	End Date	Milestone
Complete Testing for all games	November 25, 2019	<ul style="list-style-type: none"><li>- Incorporate changes on game based on pervious testing.</li><li>- Perform testing on all game.</li></ul>
User Testing	November 27, 2019	<ul style="list-style-type: none"><li>- Test final version of application with students and faculty members and people that have tremors and/or elderly people.</li></ul>
Testing Version III	November 29, 2019	<ul style="list-style-type: none"><li>- Perform final unit testing for all features and integration test.</li><li>- Perform final system testing</li></ul>

## User Acceptance Testing

User acceptance testing will be considered as a part of our development process. We plan to conduct user testing at our final version, version 3, when we have programmed “PARKinetics” to be fully functional and its games are ready to play. We will start by asking a couple of faculty members from Applied Sciences and students as well as people we personally know that may have tremors and/or are elderly to test the sample of our application from November 18<sup>th</sup>, 2019 to November 25<sup>th</sup>, 2019 from 12:00 pm - 4:00 pm each day. We will provide them with the guidelines to inform them of proper usage of the application and ask them to test all the major components of the application. Also, since our application consists of three games, besides the functionality of the application, in our guideline, we will provide a full description of the purpose of the application and how each key feature in this app is expected to work.

Our goal in this stage is to check whether our iOS application is easy to understand and easy to use. We want them to access the different layouts of our application without us helping them such as accessing the single user profile, switching between games and seeing if they are able to view and understand their progress after each game, trying buttons to see if it is clickable and functional. This can help us to examine the application from a user point of view.

During the testing period, we will be available in person with users to answer their questions and provide them with any support if needed. All the comments and feedback/suggestions will be collected from the user and if they encounter any glitches or bugs it will be our priority to fix and

then implement their feedback immediately after the user testing period has completed (after November 18<sup>th</sup>). Our goal is to complete the project before the deadline on December 2<sup>nd</sup>.

## Integration Testing

### Testing Strategy

The integration testing aims to expose flaws between the connected models of the “PARKinetics” app. Errors will be counted and recorded in 2 areas:

1. Unexpected behaviour occurs with the intended use of the application
2. The system does not degrade or crash when unexpected errors and exits occur

The test will be considered successful when the result of both the above statements is false.

The test will be Blackbox tests and test will be updated as the software is developed.

### Test Cases

Table 5: Integration Testing (Test Cases)

Tests	Successful if...
Enter a single game, and exit back to the main screen without playing a game	<ul style="list-style-type: none"><li>• No database entry is created</li><li>• Does not crash after 100 repetitions</li></ul>
Play a game, but exit after only 3 inputs (without completing games)	<ul style="list-style-type: none"><li>• Database entry is created, but initialized to zero, or entry is not created</li><li>• Does not crash upon exit</li></ul>
Play a game to completion	<ul style="list-style-type: none"><li>• New database entry is completed with relevant values</li><li>• Game exits successfully without crashing</li></ul>
Create a user profile	<ul style="list-style-type: none"><li>• New table created in the database</li></ul>
Check profile	<ul style="list-style-type: none"><li>• Radial graph generated matching averages shown in database</li><li>• Program does not crash</li></ul>

## Software Tools for Measuring Complexity

To measure the size and complexity of our code, we will be using the features that XCode provides to measure performance and capture code metrics (number of lines of code, number of classes etc.) [2].

## Quality Assurance

We have and will continue to reach out to people we know that may have personal experience with Parkinson’s Disease or with a PD patient. By doing this, we can collect more information to ensure that the movements being done in our therapeutic games will benefit and improve the patients’ symptoms. We have also reached out to *NeuroFit BC* [3], an organization that specializes in exercise-based physiotherapy for Parkinson’s and Stroke patients. We have

contacted one of their physiotherapists who will get back to us with more information and tips regarding useful exercises and key points to keep in mind when creating routines.

## References

- [1] L. Shao, "Top 5 iOS Test Automation Frameworks with Examples," Bitbar, 2019. [Online]. Available: <https://bitbar.com/blog/top-5-ios-testing-frameworks-with-examples/>. [Accessed 12 October 2019].
- [2] Apple Inc., "XCTest," Apple, 18 September 2013. [Online]. Available: <https://developer.apple.com/documentation/xctest>. [Accessed 15 October 2019].