KASPER – ID5 PROJECT DOCUMENTATION

SOFTWARE DEVELOPMENT TEAM:

Project Manager: Tushita Patel

Dev Lead: Kristof Mercier, Dylan Prefontaine

Test Lead: Jeremy Liau

Build Manager: Christopher Mykota-Reid (ChrisMR)

Developers: Gaurav Arora, Haotian (Justin) Ma, Melody (Tian) Zhao

Test Team: Christopher May (Chris May), Ryan Tetland

Documentation: Arianne Butler

Contents

[1.0 Requirements Document ID5 2](#_Toc478944984)

[1.3 Mini Milestones for ID5 3](#_Toc478944985)

[2.0 Time Estimations 3](#_Toc478944986)

[3.0 Design 4](#_Toc478944987)

[3.1 API Document 4](#_Toc478944988)

[4.0 Testing Document 4](#_Toc478944989)

[6.0 Build Report 5](#_Toc478944990)

[7.0 Incomplete Requirements 7](#_Toc478944991)

# Requirements Document ID5

* Replace descriptive text with icons (move to future)
* Thorough review of test cases for all back-end code

## 1.3 Mini Milestones for ID5

**Development**:

**Testing**:

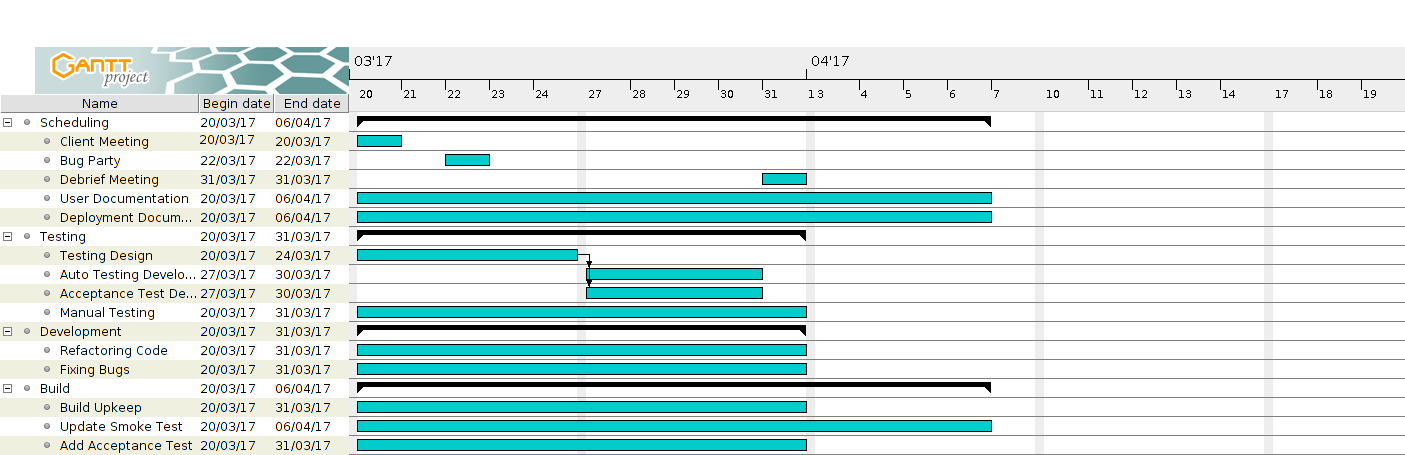
**Build**:

**Documentation**:

* Organize, compile, and edit all ID5 documentation ✔

# 2.0 Time Estimations

Gantt Diagram ID5:



To see time estimations on all individual tasks, please follow the link in Process Documentation, section 2.0 Activity Log, and click on the bottom tab labelled “Individual Activity Log”.

In ID3, development time estimations have been added for each assigned task on Trello. Trello cards now contain a coloured label indicating the approximate estimation of each task. These estimates were approximated by the dev team.

The following image shows the colour coded time estimation scale with which each trello card will be rated.



# 3.0 Design

## 3.1 API Document

The API document outlines the client server communication of our system. Our project will contain eighteen API calls, eighteen of which are currently implemented, all of which are in full communication with the server. These calls send a JSON body with relevant data. The server receives, parses, and processes the JSON using the database. When finished, the server will reply to the sender with a token containing the reply status and a JSON body with the requested information. The following link contains our updated ID4 API document and details on the various calls to our database.

Note: We are currently experiencing difficulty with this link. Please copy and paste it into your web browser.

[https://docs.google.com/document/d/1N4jt1\_PgxPhXwdc1TcT7TBjFNOZqYO5L10ha3bpO5M8/edit#heading=h.1sskatsa28we](https://docs.google.com/document/d/1N4jt1_PgxPhXwdc1TcT7TBjFNOZqYO5L10ha3bpO5M8/edit%23heading=h.1sskatsa28we)

# 4.0 Testing Document

* Jeremy
* Hallway testing write-up
* Find and fix bugs with the dev team
* Document:
  + Defect report
  + State transition diagrams
  + Test matrix
* Test team mini milestones:
  + The second bug party has been planned for March 22 nd
  + Hallway testing is planned for March 29 th
  + Regression tests will be updated to ensure no bugs are re-introduced
  + Polishing the application will be a high priority for this ID. Our goal is to eliminate as many bugs as possible, starting with the most critical ones
  + Acceptance testing

# 6.0 Build Report

Smoke Test Status:

Front-end smoke tests are implemented and passing using protractor. Further regression tests have been added to the front-end smoke test suite since ID4. Back-end smoke tests are also passing, and additional unit and regression tests have been added to accommodate new functionality from ID4.

Build Status:

The builds for iOS and Android are compiling. The Linux build and the remote server are undergoing smoke tests. After successful builds, the remote server updates from the repository. The test server is emptied before running the front-end smoke tests (is this good???). We are running end-to-end smoke tests using Protractor on the Firefox browser and testing the back-end with Python scripts. After running the smoke tests, a file with the percentage of test coverage is pushed to the server. All testing has been conducted on the browser version of our app; smoke tests are not in place for other platforms. When the smoke tests pass, the build is pushing the source code to a branch dedicated to more rigorous testing (what branch???).

SDKs, Packages, and Tools:

* Cordova CLI: 6.5.0
* Ionic Framework Version: 2.0.0-rc.5
* Ionic CLI Version: 2.2.1
* Ionic App Lib Version: 2.2.0
* Ionic App Scripts Version: 1.0.0
* npm: 3.10.10
* jdk: 1.8.0\_121
* nvm: 0.32.0
* node: 6.9.4
* Java SDK 8
* packages listed in package.json
* plus ~400 other Ionic dependency packages
* Android:
* SDK Platform Android 7.1.1, API 25, revision 3
* Android SDK Tools, revision 25.2.5
* Android SDK Build-tools, revision 25.0.1
* Android SDK Platform-tools, revision 25.0.3
* Google Repository, revision 42
* Android Support Repository, revision 42
* iOS:
  + OS: OS X El Capitan
  + Xcode version: Xcode 7.3.1 Build version 7D1014Server:
* Browser:
  + OS: Ubuntu 14.04 (trusty)
* Server:
  + Python 2.7
  + Google Cloud SDK v143.0.1
  + Python Extension for google cloud v1.9.50
  + Python Extension (Extra Libs) v1.9.49
  + Gulp v3.9.1
  + OpenSSH v6.6.1p1
  + Sshpass v1.05
* End-to-End Tests
  + Protractor v5.0.0
  + Firefox v47.0.1
  + Selenium v3.1
  + Jasmine v2.5.3

Corodova 6.5.0 requires both jdk 1.8 (or higher), as well as npm v2.2.1 and node v4.0.0. We will be using the most recent version of node and npm to reduce version conflicts. Our servers are built on Google App Engine, which requires Google Cloud. The server uses Python 2.7, Python extensions for Google Cloud SDK, and the most recent version of Google Cloud SDK. Our end-to-end tests are written in Jasmine and run by Protractor, which sends to Ionic through the Selenium server. We are currently testing our system on Firefox v47.0.1, because of compatibility issues with newer versions. We are using Gulp to create a config file which allows the test team to toggle between the real and test databases. This is to prevent filling the real database with test data. We are using sshpass to login to the database server and update files. All developers and testers have been set up with the latest versions of the required tools. We are using Xcode 7.3.1 (for what???). The Ionic dependencies are extensive, and can be viewed in further detail at the following link:

[https://ionicframework.com/docs/](https://ionicframework.com/docs/%20)

Releases for our build can be found here:

<https://github.com/CMPT371Team1/Project/releases>

# 7.0 Incomplete Requirements

The following requirements will not be implemented this term. Each will be documented and passed on to future programmers in our Programmer Documentation.

* Sign in / Sign up using Facebook
  + This feature is incomplete in the front-end, but complete in the back-end.
* Messaging Functionality­
  + This feature is incomplete in the front-end, but complete in the back-end. It includes functionality for sending initial messages to sellers, marking read messages, and deleting messages. Once implemented, it will need to be integrated with the back-end and tested.
* Edit listing
  + This feature is incomplete in the front-end, but complete in the back-end
* Push notifications
  + The app should be able to track changes in the database and trigger notifications to interested buyers
  + Sellers should receive a notification if a prospective buyer sends a message regarding one of their Listings
  + Buyers can elect receive “hot list” notifications for one or more of the following:
    - A Seller publishes a Feature Listing
    - A new Listing is added that fits some predetermined criteria (i.e. neighborhood, price range, seller, etc.)
    - Price change on a Favourited Listing
    - A Listing is published that contains similarities to past searches
  + Sellers can elect to receive notifications regarding their personal Listings:
    - A Listing has been Favourited by a user
    - A buyer sets a price watch on a Listing
* Integration with Google Maps
  + This feature should show Listings with regards to their physical location on a map. The Google Maps functionality should be accessible from the Detail page.
* Forgot password
  + This feature should allow a user to request an email or text message containing reset information
* Re-order a Listing’s images.
  + This feature should allow a seller to select a primary image to be displayed on the browse page, and to re-order a Listing’s images.
* Super Admin User
  + A Super Admin User can access the following additional functionality:
    - Publish a Listing under any user
    - Edit any Listing
    - Remove any Listing
* API Versioning
* Edit User - Check up on this nearer to deadline.