KASPER – ID4 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 ID4 3](#_Toc477599529)

[1.1 Front-end Requirements 3](#_Toc477599530)

[1.2 Back-end Requirements 3](#_Toc477599531)

[1.3 Mini Milestones for ID4 3](#_Toc477599532)

[2.0 Time Estimations 4](#_Toc477599533)

[3.0 Design 5](#_Toc477599534)

[3.1 API Document 5](#_Toc477599535)

[3.2 Development Unit Tests 5](#_Toc477599536)

[4.0 Testing Document 6](#_Toc477599537)

[5.0 Coding Style Guide 6](#_Toc477599538)

[6.0 Build Report 6](#_Toc477599539)

[7.0 Upcoming Requirements 8](#_Toc477599540)

[7.1 ID5 Requirements (Polishing and Quality Assurance) 8](#_Toc477599541)

[7.2 Incomplete Requirements 9](#_Toc477599542)

[8.0 Triage Meeting ID4 9](#_Toc477599543)

# Requirements Document ID4

## 1.1 Front-end Requirements

* Form control validation for all remaining pages (check back closer to deadline)
* Fix description box in Add Listings (done)
* Display currently added images during upload image process (done)
* Make Favourites page the same style as Browse page (done)
* Get front-end fully integrated with back-end (done, except for the messages, which were requested late by the client, and only put in the backend)
* Finalize and complete UI flow (done)

## 1.2 Back-end Requirements

* Email verification
* Logout
* Contact Seller (done in back end only)
* Sign in / Sign up using Facebook (done in back-end only)
* Delete Listing
* Get the front-end fully integrated with the back-end for all API calls (done except message API)
* Get messages/edit messages/contact seller (done in the back-end only)

## 1.3 Mini Milestones for ID4

**Development**:

* Kristof

**Testing**:

* Do Peer Review board tasks (as many as possible) ✔
* Manual testing (Browser, Android) ✔
* Remove unused and out-of-date test files ✔
* Automate regression tests ✔
* Create separate configuration files for Firefox and Chrome testing ✔
* Get backend developers to create a test database for testing ✔
* Fix high priority bugs before Bug Party ✔
* Create new Use Cases for ID4, update old ones ✔
* Refactor and update E2E tests ✔
* Testing results and defect report ✔
* Fix smoke tests ✔
* Test matrix update ✔
* Path diagram update ✔
* Testing plan document update ✔
* Hallway Testing ✘
* Acceptance Testing ✘

**Build**:

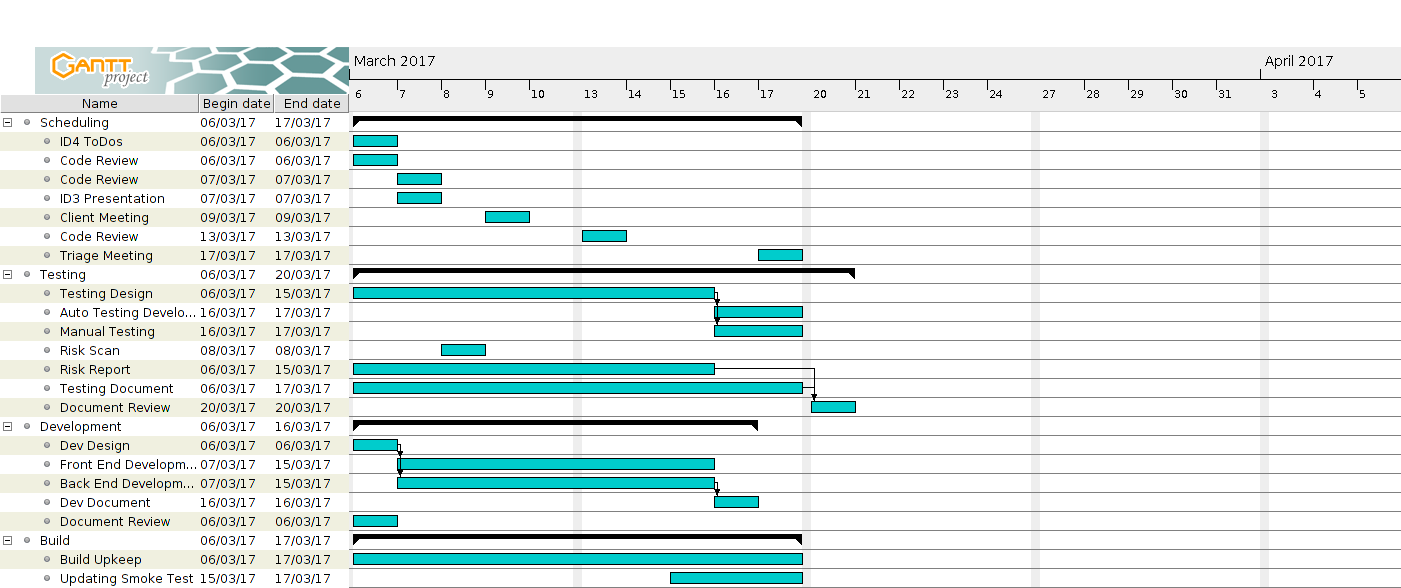
* Get broken smoke tests working ✔
* Integrate test server with the build ✔
* Make the build more automated ✔
* Add GitHub release automation ✘
* Add style checking ✘

**Documentation**:

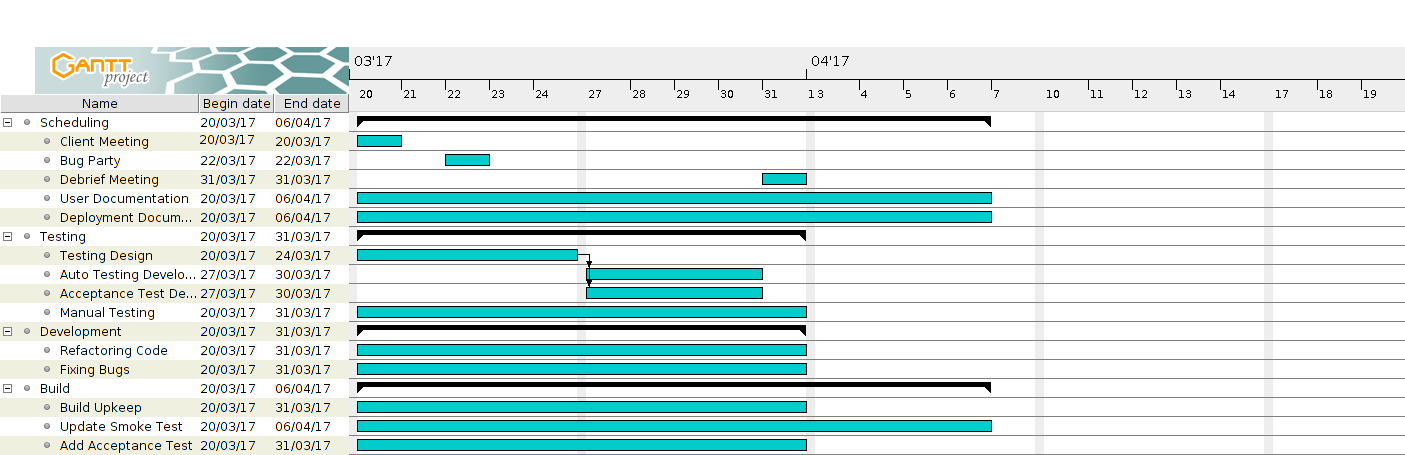
* Organize, compile, and edit all ID4 documentation ✔

# 2.0 Time Estimations

Gantt Diagram ID4:



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 (calls to the server), eleven of which are currently implemented, and two 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)

## 3.2 Development Unit Tests

The procedure to be followed by all developers for writing both front and back-end unit tests respectively, can be found at the following links:

Front-end Guide for Unit Testing:

<https://github.com/CMPT371Team1/Project/wiki/Coding-Style-Example-(JavaScript)>

Back-end Guide for Unit Testing:

<https://github.com/CMPT371Team1/Project/wiki/Coding-Style-Guide-(Python)>

# 4.0 Testing Document

* Jeremy

# 5.0 Coding Style Guide

The development team has put together a set of guidelines to serve our purposes. These guidelines highlight the salient features of the coding style to be followed by developers. Useful examples are provided for quick referencing.

For front-end development using Ionic, the coding guide and sample can be found at the following wiki page:

<https://github.com/CMPT371Team1/Project/wiki/Coding-Style-Example-(JavaScript)>

For back-end development using Python, the coding guide and sample can be found at the following wiki page:

<https://github.com/CMPT371Team1/Project/wiki/Coding-Style-Guide-(Python)>

# 6.0 Build Report

Smoke Test Status:

Front-end tests are implemented using protractor and are currently passing. Back-end smoke tests are also implemented and passing.

Build Status:

The builds for iOS and Android are undergoing light smoke tests, ensuring compilation. We are experiencing difficulty with setting up the build for these platforms, and are focusing on other priorities before the code freeze. Both the Linux build and the remote server are undergoing smoke tests. The server is being updated from our repository after successful builds. Integration of the test server was implemented in ID4, which involves refreshing the test server before running the end-to-end tests, as well as updating after successful builds. End-to-end smoke tests are being conducted using Protractor on the Firefox browser. The back-end is being tested with Python scripts. Thus far, all testing has been done on the browser version of our app; smoke tests are currently not in place for other platforms. Upon successful completion of the smoke tests, the build should push the source code to a branch dedicated to more rigorous forms of manual testing.

SDKs, Packages, and Tools:

All SDKs, packages, and tools employed in our build, as well as their version number, are subject to change. These frameworks are still in question due to lack of experience. These decisions will be made final once the build manager has a firm understanding of automated testing, deployment, server builds, and system builds.

Current list of SDK’s, 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
* 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:
* 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. Google Cloud is required for the Google App engine; the platform our servers are built upon. The server uses Python 2.7, the most recent version of Google Cloud SDK, and Python extensions for Google Cloud SDK. Our end-to-end tests are driven by Protractor, which sends to Ionic through the Selenium server and is driven by Jasmine. 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 sets the database of our app to the test database, instead of the actual one, if the app is undergoing tests. This is to prevent filling the actual database with test data. Sshpass is used to provide a password and ssh on the command line, allowing us to automatically login to the database server and update the 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 the time being. 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/tree/develop/releases/ID3>

# 7.0 Upcoming Requirements

## 7.1 ID5 Requirements (Polishing and Quality Assurance)

* Replace descriptive text with icons (check back closer to deadline)
* Get a legitimate Kasper icon (check back closer to deadline)
* Thorough review of test cases for all back-end code (TODO in ID5)

## 7.2 Incomplete Requirements

This section outlines unfinished requirements. These requirements will not be implemented this term and will be documented and passed on to future programmers in Programmer Documentation.

* Contact Seller (front-end only)
* Sign in / Sign up using Facebook (front-end only)
* Get messages/edit messages/contact seller (front-end only)
* Push notifications if something changes regarding a Listing saved in Favourites (a change in the database triggers a notification to the User’s who have favourited that Listing)
* Integration with Google Maps (future)
* Add Google Maps section to Detail page (removed – future)
* Forgot password (future)
* Implement the message system in the front-end and integrate it with the backend (removed – future)
* Finalize method of ordering images
* Send a message to the seller including contact information
* Users can sign up to receive “hot list” notifications
  + Feature Listing (paid for by Sellers)
  + Newly added Listings
  + Price changes on Favourites
  + Based on previous search history
* Sellers receive notifications regarding personal Listings:
  + When a Listing is saved to Favourites
  + When someone requests a viewing
  + When someone sets a price watch
* Super Admin User
  + Log in as Super Admin
  + Add new Listing under any user
  + Edit any Listing
  + Remove any Listing

# 8.0 Triage Meeting ID4

**Date**: Friday, March 17th /17

**Start Time**: 1:30pm

**End Time**: 2:25pm

**Location**: Spinks320

**Members Present**: Tushita Patel (Project Lead), Kristof Mercier (Dev lead), Jeremy Liau (Test Lead), Chris Mykota-Reid (Build Manager)

**Development:**

All Trello tasks have been implemented, except for:

1. Google maps integration with front-end and back-end (but is setup for the database)
2. Buyer to Seller messaging on front-end (this API is implemented in the backend)
3. “Forgot Password” feature in both front and back-end

Due to the ID5 code freeze, the team has decided to document and abandon these requirements in favour of producing better quality code for the finished areas. The above requirements will not be delivered to the client.

**Testing:**

1. All ID4 tests will be performed (manual, end-to-end, etc.)
2. The test team has finalized plans for acceptance and hallway testing

**Bugs / Defects:**High priority bugs to be fixed by the end of ID4 (Sunday, March 19th, /17)

Medium priority bugs to be fixed by the Bug Party (Wednesday, March 22nd, /17)

Low priority bugs will be fixed by the end of ID5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **STI** | **Decision** | **Severity** | **Priority** |
| 1. | Bathroom filter stops functionality of the other filters | This STI is probably appearing due to lack of back-end testing. It could be tricky to fix. Fixing this bug should be postponed until ID5. | Low | Low |
| 2. | Liking and Disliking while not signed in returns a runtime error on details page | Although fixing this bug is important, we estimate that fixing it will be too time consuming to prioritize during ID4. | High | Low |
| 3. | When a user is not signed in and they attempt to swipe right or left on a Listing, the Listing turns red or green | This is a front-end bug and could be easily fixed. It is a frequently occurring bug that should be prioritized. | High | High |
| 4. | Save button and Sign-out button on My Profile page are not working while signed-in. | Although fixing this bug is of high importance, this STI could be postponed to early next week because there are other, higher priority bugs. | High | Medium |
| 5. | Sign-in using Facebook is not implemented on the front-end. | This issue might not be too tricky to fix, but is important to the client. We should consider this bug of medium priority. | Low | Medium |
| 6. | Browse page does not add more Listings to the cycle when you Like/Dislike a Listing while signed in | This is an important issue because it is a major feature. It should be fixed as soon as possible. | High | High |
| 7. | No refresh after every sign in. | Might be an easy fix. Not an important issue, comparatively. | Low | Low |
| 8. | Disliking on the Browse Page still moves the Listing to Favourites. | Might be an easy fix. Although severe, it is of lower priority, comparatively. | High | Medium |
| 9. | Add Listing is unable to publish Listings. | Might be the same problem as postal code. Quite severe. Should be fixed as soon as possible. | High | High |
| 10. | There should be a “required image” pop-up when attempting to publish a Listing without at least one image. Currently not displayed. | Might be an easy fix. Not too sever. | Low | Low |
| 11. | My Listings and My Favourites page should have an Edit button on the top right of the app | Swiping left to get Delete/Edit options for a listing is not intuitive. This is a potential design change. Not a priority right now. | Low | Low |
| 12. | “Missing Postal code” even when it’s been added. | Fixing of this STI depends on 9. This may be the same bug as 9, or a bug resulting from 9. This bug will be tracked after 9 is fixed. | Low | Unknown |
| 13. | Cannot remove Listings from Favourites list. Reappears. | Might be the same bug as 8, or a bug being caused by 8. | High | Medium |
| 14. | When registering a user on android, the text fields scroll all the way up. | This bug was identified at the end of ID and has not yet been fixed. It should be prioritized soon. | High | Medium |

**Build:**

1. Troubleshooting issues and regressions
2. Added test servers to implementation
3. Reset the test servers before running end-to-end smoke tests on the front-end
4. Shared build knowledge with Gaurav during pair programming sessions
5. Chrome smoke tests removed due to increasing complexity and bugs
6. Added installation of Jasmine Spec Reporter to the build
7. Added if-statements to the build for less manual modifications on release
8. Still looking into automated Style Checking and automated GitHub releases
9. updated