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 (ChrisJ), Ryan Tetland

Documentation: Arianne Butler

Contents

[1.0 Requirements Document ID4 3](#_Toc477208956)

[1.1 Front-end Requirements 3](#_Toc477208957)

[1.2 Back-end Requirements 3](#_Toc477208958)

[1.2 Mini Milestones for ID4 5](#_Toc477208959)

[2.0 Time Estimations 5](#_Toc477208960)

[3.0 Design 6](#_Toc477208961)

[3.1 API Document 6](#_Toc477208962)

[3.2 Updated Data Structures ID4 ??? 6](#_Toc477208963)

[3.3 Development Unit Tests 7](#_Toc477208964)

[4.0 Testing Document 7](#_Toc477208965)

[5.0 Coding Style Guide 7](#_Toc477208966)

[6.0 Build Report 8](#_Toc477208967)

[7.0 Upcoming Requirements 8](#_Toc477208968)

[7.1 ID5 Requirements 8](#_Toc477208969)

[7.2 Future Requirements 8](#_Toc477208970)

[8.0 Triage Meeting ID4 9](#_Toc477208971)

# Requirements Document ID4

## 1.1 Front-end Requirements

Priority Front-end Requirements

* Form control validation for all remaining pages
* Replace descriptive text with icons
* Fix description box in Add Listings
* Add Google Maps section to Detail page
* Display currently added images
* Get a legitimate Kasper icon
* Finalize method of ordering images
* Make Favourites page the same style as Browse page
* Get front-end fully integrated with back-end
* Finalize and complete UI flow

Front-end Requirements Changes

This section highlights the changes to our UI since ID3. These changes were discussed and agreed upon during both the current and previous ID’s, and have been implemented during ID4.

Finalized changes to the UI for ID4 can be found at the following link: ???

## 1.2 Back-end Requirements

Priority Back-end Requirements

* Email verification (not done for ID3)
* Forgot password (not done for ID3)
* Confirm email
* Logout
* Contact Seller
* Sign in / Sign up using Facebook
* Delete Listing
* Get the front-end fully integrated with the back-end for all API calls

Other Back-end Requirements

* Thorough review of test cases for all back-end code
* Push notifications if something changes regarding a Listing saved in Favourites (a change in the database triggers a notification)
* Integration with Google Maps

Do we need this still???

**System Design:**

The back-end system implementation is separated into two main modules – User accounts and Listings information. The account module handles user Sign-in, Sign-out, Sign-up, email verification, forgotten passwords, and resetting passwords. The Listings module defines a set of data related to a listed property, such as its location, price, description, images, etc. It includes creating new Listings, getting filtered Listings, and edit existing Listings. To start the back-end, the http server is initialized, which calls all system modules before serving user requests. Thus, it has complete control over all parts of the system, and can decide to close any aspect should an issue arise.

**System Requirements:**

Fundamental aspects of the back-end behaviour can be defined by the following set of requirements:

**Functional Requirements:**

1. The back-end must gather data sent from devices and store it in the database for future reference.
2. User requests must be handled appropriately, and relevant information stored in the database must be sent to the device interface for display.
3. The system should be capable of recovering from failures and crashes whilst maintaining the integrity of any stored data.

**Non-functional Requirements:**

1. The back-end system should be responsive to user requests, so that delays in displaying data are minimized.
2. Data integrity and error correction mechanisms should be implemented so that no erroneous data is stored in the database.
3. The system should send informative error messages to the client about the source of error.
4. The system should provide an appropriate debugging environment, in which new code can be easily integrated, tested, and checked for errors.

**Software:**

The back-end system is implemented in Python and uses several external sources for specific implementations:

1. Google App Engine
2. NoSQL
3. Google Datastore NDB Client Library
4. Webapp2: a lightweight Python web framework

Back-end Requirements Changes

This section highlights the changes to our back-end since ID3. These changes were discussed and agreed upon during both the current and previous ID’s, and have been implemented during ID4.

## 1.2 Mini Milestones for ID4

**Development**:

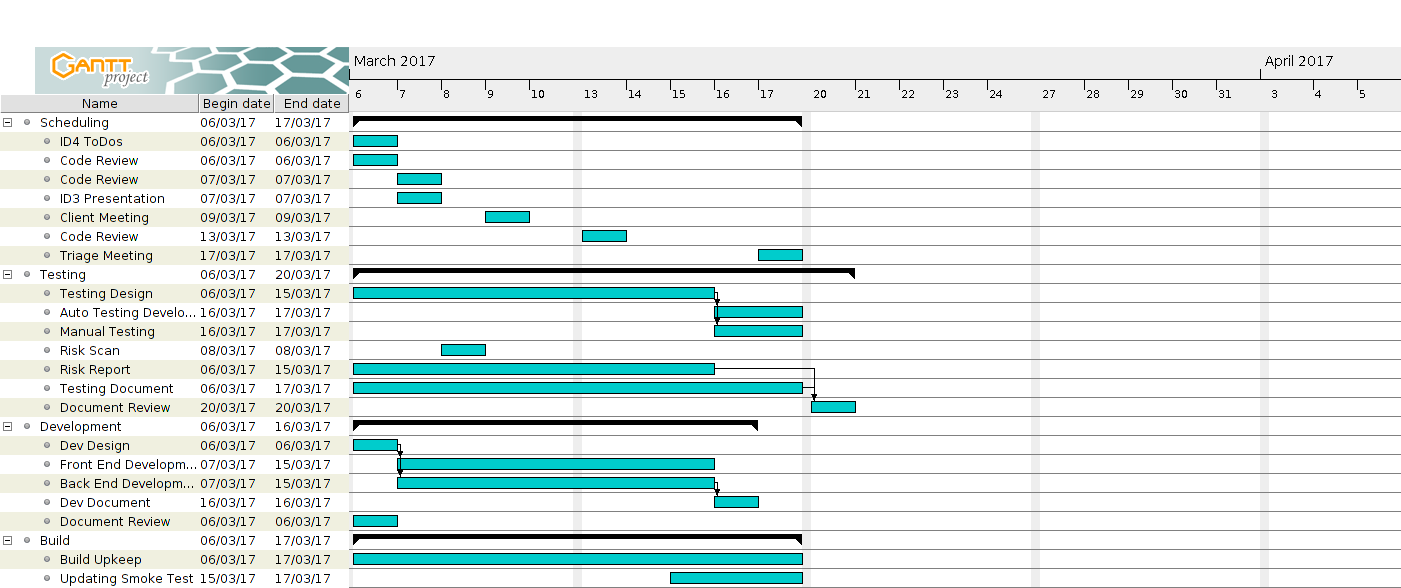
**Testing**:

**Build**:

**Documentation**:

# 2.0 Time Estimations

Gantt Diagram ID4:



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 sixteen 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, it will reply to the sender with a token containing the reply status and a JSON body containing 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 Updated Data Structures ID4 ???

## 3.3 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

* Chris MR

# 7.0 Upcoming Requirements

## 7.1 ID5 Requirements

## 7.2 Future Requirements

This section outlines requirements identified as “nice-to-have’s” for upcoming ID’s. Certain client requirements have been identified as too ambitions for the allotted time frame, and are not expected to be completed.

* 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**:

**Start Time**:

**End Time**:

**Location**:

**Members Present**:

**Summary**:

1. **Front-end:**
2. **Back-end**
3. **Testing**
4. **Build**