*Candid —*

SE452 – Object Oriented Enterprise Application Development

Enabling continuous and constructive feedback for teams

Contents

[Overview 2](#_Toc525745109)

[Requirements 2](#_Toc525745110)

[Epics & User Stories 2](#_Toc525745111)

[Description of problem 3](#_Toc525745112)

[Design 3](#_Toc525745113)

[Discussion of how your design met the requirements 3](#_Toc525745114)

[Discussion of lessons learned 3](#_Toc525745115)

[Milestone Screenshots 4](#_Toc525745116)

[Decision Log 5](#_Toc525745117)

# Overview

A web application that allows teams to request and provide continuous feedback for each other.

# Requirements

## Epics & User Stories

*Users should be able to give and receive feedback.*

* As a user, I want to give feedback to another user, so that I can give that person feedback to help them grow.
* As a user, I want to give feedback to another user anonymously, so that I am more likely to give more candid and honest feedback.
* As a user, I want to view feedback from another user, so that I can make improvements based on their feedback.
* As a user, I want to view all of the feedback I have received, so that I can see how my feedback has changed over time.
* As a user, I want to view all of the feedback I have given, so that I can see how the feedback I’ve given has evolved.
* As a user, I want to request feedback from specific users, so that I can get feedback from people that are important to me.
* As a user, I want to know that the feedback I gave was read, so that I know if that person is reading their feedback.
* As a user, I want to search by specific terms to find relevant feedback, so that I can find old feedback to track specific themes.
* As a user, I want to reply to feedback that I have received, even if it was given anonymously, so that I can ask questions and gain clarification.
* As a user, I want to be able to reply to feedback replies, so that I can answer questions from the person whom I gave feedback.

*Users should be able to create and save feedback templates and create feedback request recurrences.*

* As a user, I want to create templates with specific questions when requesting feedback, so that I can point the conversation toward specific themes.
* As a user, I want to save feedback templates, so that I can reuse a format that I like when requesting feedback.
* As a user, I want to share the templates I have created with other users, so that there can be standardized formats available for others.

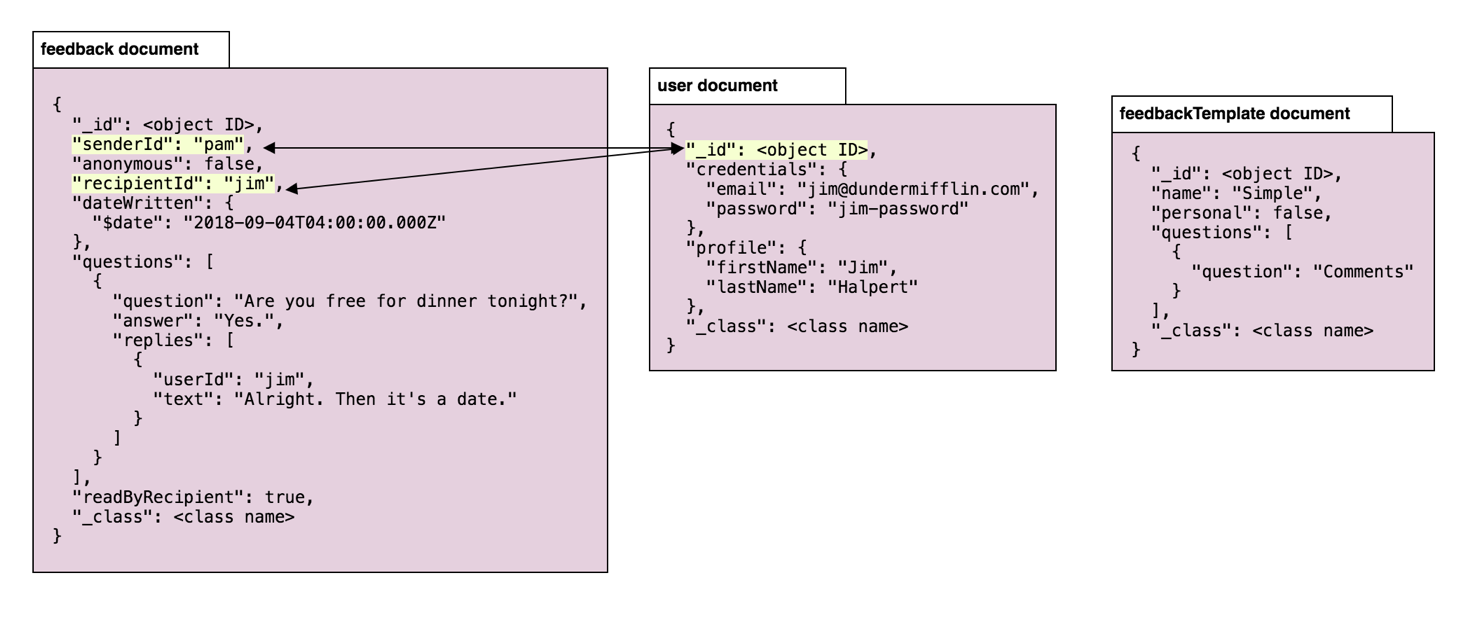
*Users should be able to manage their public and private data.*

* As a user, I want to change my email address, so that I don’t have to create a new account if my email address changes.
* As a user, I want to change my password, so that I can ensure my account is protected.
* As a user, I want to manage a basic profile which includes my full name, so that I can be found by name by other users.
* As a user, I want to include some basic information about myself in my profile, so that I can write in freeform anything about myself for others to see.
* As a user, I want to upload a profile picture so that other users can see a picture of me.

## Description of problem

Teams of people want a way to provide formal feedback to each other in a convenient manner. This application should allow teams to effortlessly interact candidly via feedback.

# Design



# Discussion of how your design met the requirements

## Feedback Document

This document is designed to enable basic feedback (question/answer), allow anonymity (anonymous flag), ability to know if it has been read (readByRecipient flag), and creates the option to write replies (each question object has an optional replies array associated, which is recursive, allowing a full thread).

## FeedbackTemplate Document

Feedback Templates will be used heavily by the UI. Users can create templates, choose to allow them to public (to enable using another users templates), and create feedback (and requests) from an existing template

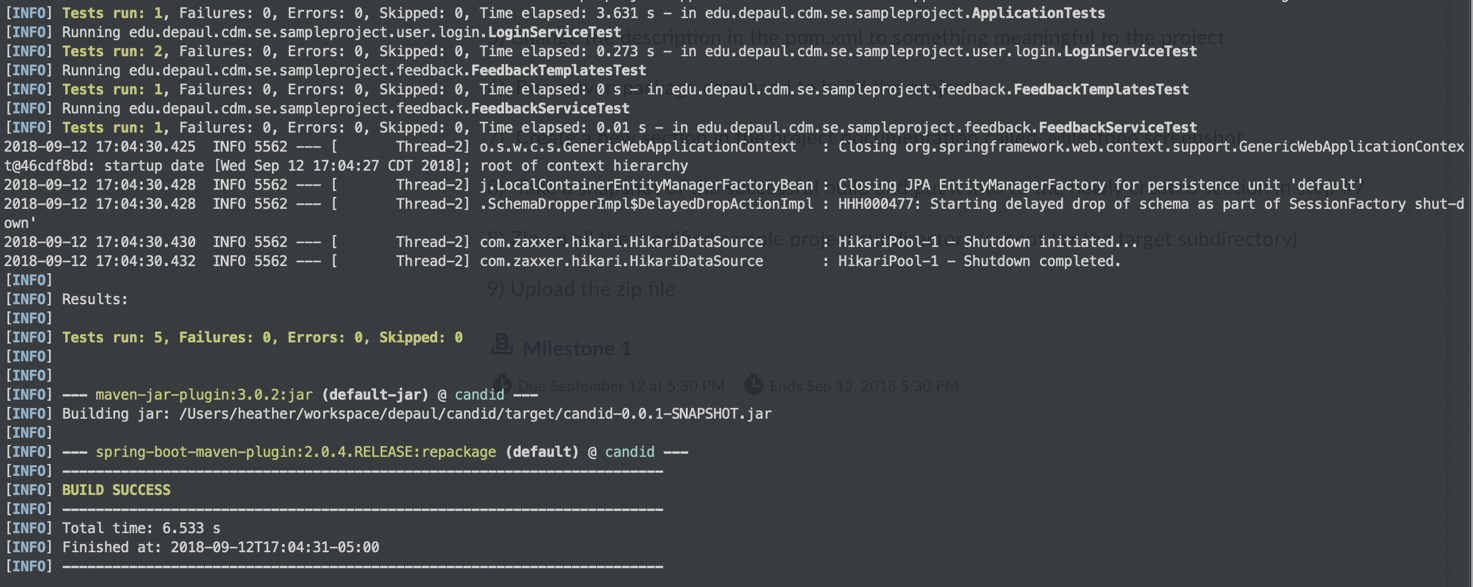
## User Document

The user document exists to maintain credentials and profiles. Users can update any data on their own user object (excluding object ID). When the user logs in, the User object is turned into an AuthorizedUser object with all the necessary fields for the UI (this keeps the password field out only in the LoginService and nowhere else).

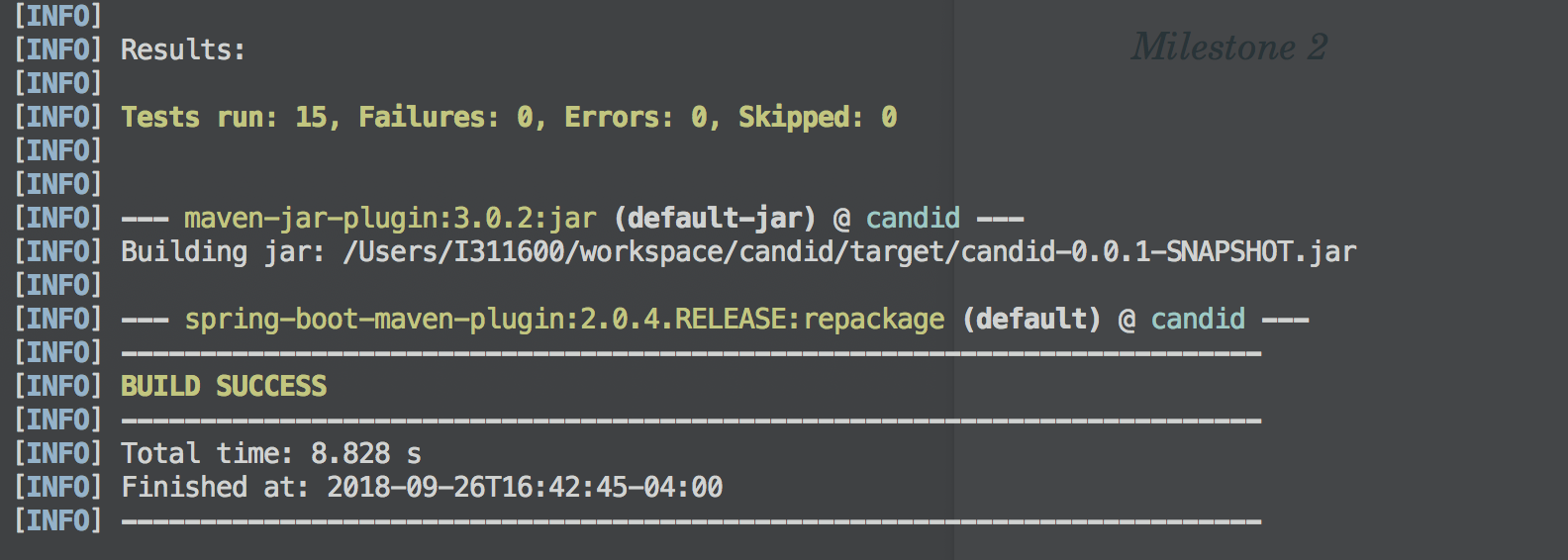
# Discussion of lessons learned

# Milestone Screenshots

#### Milestone 1



#### Milestone 2



# Decision Log

|  |  |  |  |
| --- | --- | --- | --- |
| **Problem** | **What was decided** | **Alternatives considered** | **Rationale** |
| Which IDE to use | IntelliJ | None | I have a license for IntelliJ and have high familiarity with the IDE. |
| Code repository | Github | None | It’s better to have it somewhere centralized than nowhere at all. |
| Extra features | User Profile | None | The professor asked me to include another feature, so I am adding the ability for the user to manage a public profile, and also manage their private information (email, password). |
| Which DB platform to use | MongoDB | Neo4j | I considered Neo4j, as this application stores feedback (relationships) between people, but I ultimately decided to go with MongoDB because the objects are relatively unchanging and may be simpler to manage in document format. Feedback can be portrayed as relationships but will be easier to maintain as documents. |