CS3300: Introduction to Software Engineering

Project Plan for CS 3300 Messaging App

Version 0.0: 5-28-2015

Team: API Team

Table of Contents

IMPORTANT:

The paragraphs written in the "Comment" style are for the benefit of the person writing the document and should be removed before the document is finalized.

Also, don't forget to update the Table of contents before submission. (Right Click -> Update Field)

1

Introduction

1.1 Overview (Executive Summary)

The objective of this project is to host an API for a messaging client. Our clients will use this web service to send and store messages. They will depend on our API for managing both public and private chat groups. Our system will also the support the use of bots for commands, translations, and other tasks.

1.2 Definitions and Acronyms

List any project definitions and acronyms introduced to the project by this plan. Do not repeat items covered in a global project definitions and acronyms document. Be sure and define all acronyms before their use. This section can establish a common project vocabulary so everyone on the project understands technical terms that will be used.

HTTP: Hypertext transfer Protocol
REST: Service that accepts http requests (Representational State Transfer)
GET: Requests a resource from the server. (Web page or image)
POST: Similar to get but places additional info is sent in the body of the request.
PUT: Upload a resource to the server
DELETE: Delete a resource from the server
Spring: Java Developmental Framework
IntelliJ: Interactive development environment (Will be used with Spring)
Maven: Software project management tool
Jackson https://github.com/FasterXML/jackson : Java JSON parsing library
URI: Uniform Resource Identifier
status codes (Ex: 404, 500) - responses to network requests
JSON: Javascript Object Notation.
API: Application programming interface. Specifies how software components should interact
Slack: The communication method set by the team. This application will help us organize team tasks and documents.
GroupMe: The communication method set by the team. GroupMe is an application available on mobile and web platforms.

2 Management Structure

This section covers all aspects of managing the project that are not related to estimating, planning, and controlling the work.

1.1 Project Lifecycle

As the API team we need to have a clear set of requirements and documentation from the start for other teams to utilize. Since the functionality of our system must remain constant and is unlikely to change throughout the project's lifecycle we will use a modified Waterfall cycle.

Requirements analysis and definition:

- Document the necessary POST functions that will be used by other teams

System and software design
- Create database schema
- Produce and consume JSON responses and requests

Implementation and unit testing
- Provide a method for external clients to authenticate users
- Create unit tests that are consistent with our documentation

Integration and system testing
- Use unit tests to verify system
- Give API access to our external clients

Operation and maintenance
- Monitor performance as the app is used
- Create, or link from another service, a bug reporting module for unexpected run time reports

1.2 Roles and Responsibilities

Role	Responsibility
Project Manager	Magnus
Planning and Tracking Lead	Sam
Requirements Lead	Andrew
Design Lead	Jesse
Implementation Lead	Magnus
Quality Assurance Lead	Sam
Development Engineers	Andrew
QA Engineers	Jesse

1.3 Communication

We will use the project management tool Slack to discuss the project and share relevant links and files. We will use the phone application GroupMe for instant communication to all team members. Google Drive will used be for collaborating on files simultaneously.

We will meet in person only as needed. These meeting times will typically be held on Mondays or Wednesdays at 7pm in the CoC Commons.

2 Risk Management

1.1 Risk Identification

Risk	Probability	Severity	Description

Unclear API for other teams usage	High	High	Other teams need a clear and complete definition of our API's functions early on in the lifecycle
Authenticating our clients	Medium	Medium	How will we authenticate all our different clients?
Unexpected REST requests and responses	Low	Medium	How will we handle unexpected API requests and insure there are no unexpected responses based off our documentation?
Bot integration completion	Medium	Low	Bot integration and testing could be too time consuming for the allotted time
Database Schema is not ideal for all teams	Medium	Medium	The database will be accessed by three different teams on two different platforms. It may be difficult to create APIs that are ideal for everyone

1.2 Mitigation Plan

Write down steps that you will take to mitigate the risks.

1. Unclear API : provide documentation early on of the expected functions and notify other teams of any necessary changes
2. Authentication : We will use Spring Security to handle authentication. It can easily added as a dependency using Maven
3. Unexpected requests/responses : We will use Spring''s server side tests to insure there are no unexpected requests/responses.
4. Bot integration : Focus on completing a single bot to use as a template to add others
5. Database Schema: Meet with every team and use their requirements and recommendations to create the database

2 Planning and Control

This section covers all aspects of managing the project related to planning, and controlling the work.

1.1 Milestones

This section lists key milestones for the projects. Milestones should have a name and a due date. Milestones are NOT tasks. Tasks are covered in the next section. Milestones consume no resources or time, they are like binary yes/no checkpoints for the project. Ideally, you should have a milestone a week for the project. That way you know you are never more than a week behind. Some milestones are given to you, like Design Document Turned-In. Others you will need to develop, like Initial Prototype Tested. You need to pick milestones that will help you track your project status.

June-02 : Requirements Doc Due
June-04 : Design Doc Due (Includes our API documentation)
June-09 : Testing Plan Due
June-11 : Host API
June-18 : All REST functions completed

June-25 : Project Due

1.2 Work Breakdown Structure

This section breaks down all work on the project into work packages or tasks. You should break down the tasks to a level so that you can assign them to a specific person, you understand the task well-enough to do it, and you can determine when you are done (the exit criteria).

- Create documentation for Messages
    - Send, Receive, Status (Read/Unread)
- Create documentation for Groups
    - Create, Join, Leave, Invite, List
- Create documentation for Documents
    - Upload, View
- Create documentation for Notifications
- Create documentation for Authentication
- Create documentation for Bots
    - EchoBot, CommandBot, BuildBot, TranslateBot
- Prototype Database Schema
- Create a public host for our API
- Create authentication for our API using Spring Security
- Complete REST Functions for Messages
- Complete REST Functions for Groups
- Complete REST Functions for Documents
- Complete REST Functions for Notifications
- Complete REST Functions for Authentication
- Complete REST Functions for Bots
- Complete server and unit tests for Messages
- Complete server and unit tests for Groups
- Complete server and unit tests for Documents
- Complete server and unit tests for Notifications
- Complete server and unit tests for Authentication
- Complete server and unit tests for Bots

1.3 Schedule

Tasks	**Prerequisites**	**Time to Complete**	**Person**	**Completed?**
Create Documentation		week 1	ALL	No
Prototype Database Schema	Create Documentation	week 1	ALL	No
Create Public Host for API	Prototype Database Schema	week 2		No
Create Authentication for API	Prototype Database Schema	week 2		No

Complete REST Functions for Messages	Prototype Database Schema, Complete REST Functions for Users	week 3,4,5		No
Complete REST Functions for Groups	Prototype Database Schema, Complete REST Functions for Users	week 3,4,5		No
Complete REST Functions for Users	Prototype Database Schema	week 3,4,5		No
Complete REST Functions for Bots	Prototype Database Schema	week 5		No

1.4 Tracking and Control

We will utilize Git to track our tasks and add details such as expected time to completion and type of task using Git's issue feature. Each team member is responsible for recording their progress on their tasks, and each member can create new tasks.

2 Technologies to be used

IDE's:	IntelliJ
OS's:	Mac OSX, Windows
Frameworks:	Spring, Maven, Gradle
Project Management:	Slack, GitHub