CAssess User Manual

14th October 2017 - In Progress

INTRODUCTION

The CAssess platform is a combined software application which aims to allow management and progress tracking of software projects (which are comprised of students in teams) by Professors, Instructors, and/or TAs. Essentially the software projects must be using one or more of 3 external tools in order to have this software utilized, they are Slack, GitHub, and Taiga.

PURPOSE

To allow the management via web interface of software project teams, usually in an academic setting, by providing real time progress data across Taiga (Agile Scrum Board), GitHub (Code Activity), and Slack (Team Communication). The users, teams, and admins are also able to be managed with this software once initialization occurs.

STRUCTURE

The CAssess core is Java and Spring based, and operates as a standalone system which is accessible via its REST API using JSON commands and data. The details of how to communicate and interact with/use it are in the “CAssess API” document.

The back end of the CAssess system is a MySQL database, which is utilized via Hibernate from the core program. Data is stored, queried, altered, etc as needed for project and user management.

The Front end of the CAssess system is HTML/HTML5 with AngularJS used to manage the dynamic interactivity to the user, and also as a controller to communicate via REST communication to the core program as needed. Typically data is sent back to the front end following a request made via REST as a result of the user accessing a page or object on a page.

Currently the server is loaded at http://cassess.fulton.asu.edu:8080/cassess

SCOPE

The scope of this project includes:

- Software Projects being managed
- Team involved in a software project
- Members/Students of a software project team
- Course to which Teams (and thereby students) are a part of
- Professors, Instructors, and TAs where applicable, to manage the members and teams

DEPLOYMENT PREREQUISITES

Web Server

Whatever platform CAssess is loaded onto, must be running a web deploying program such as Tomcat. This allows the web interface to the user to be accessible, communicating to the core CAssess program via its REST API.

Database

Currently CAssess only supports back end data interface with MySQL, thus the platform on which CAssess is loaded will also have to include a running instance of the MySQL database on it.

OS Compatibility

Currently CAssess can be loaded on Linux or Windows, it has been thoroughly tested on Linux RHEL (Redhat Enterprise Linux), and Windows 7.

Browser Compatibility

CAssess has been thoroughly tested for front end access with all current versions of Chrome, Firefox, Internet Explorer, and Safari.

Note: Access via SSL is available and has been tested successfully, but has been disabled for the use on its current server deployment.

INITIALIZATION

In order to begin managing a new software project or set of projects, a set of new data must be provisioned to the CAssess REST API. Information on how to accomplish this is found in the related “CAssess API” document under “Course Package - Create”, once a new course is created, admins, students, teams, and projects can be added/removed/edited as needed (this information is also in the CAssess APi document). You must ensure that any used emails, tokens, IDs (Github and Slack), or Taiga slugs are actually live and existing associated values, otherwise the data gathered and tracked will of course be incorrect.

USER ROLES

REST User

The REST user is only able to access the externally available CAssess REST API, utilizing any commands other than user management listed in the CAssess APi document.

Super User

The Super User is able to access all of the metrics for all courses, teams, and members of a software project. They are also able to add/alter/remove any other users, as well as courses/teams/members.

Student User

The Student user is able to access top level graphical metrics for the course(s) they are in, as well as detailed metrics for the team they are in, and their own personal progress of the software project(s).

Admin User

The Admin user is able to access detailed metrics for the course(s) they are part of, as well as for all teams and members of software projects in that course.

BUILD REQUIREMENTS/PARAMETERS

Deployed Format

The current output format being used is to create a WAR file for web deployment, no other means of output for this code build have been used or tested.

Development Environment

CAssess has been built mainly using JetBrains IntelliJ IDEA, however this was simply out of preference and usability, other IDEs may work just as well or better.

System/Internal Dependencies

CAssess has been tested and built with JRE/SE 8, currently all other dependencies are managed with Apache Maven (for core and back end) and Bower (for web components). Bower typically does not need to be installed manually but will occur automatically through Maven, however your IDE (if used) will need to have Maven capability and Maven 3.5.0 may need to be installed manually.

- Please be aware that the initial build from a new code base will take longer than all subsequent builds, as the dependencies are retrieved and loaded.

For a list of dependencies managed by Maven, please see the pom.xml document.

For a list of dependencies managed by Bower, please see the bower.json document.

Note: Bower requires npm, git, and node to be installed as dependencies

Third-Party/External Dependencies

For testing and using the system locally, similar to the deployment you will need an instance of MySQL running, as well as a local web deploy tool such as Apache Tomcat to load the WAR file or to load directly from your IDE for testing.

For MySQL the specific path, credentials, and instance information will need to be customized to any different instances other than the RHEL server it was designed for. These changes will need to be made in the DBConfig.java file.

For Tomcat, the only requirement is that port 8080 is currently used, but this can be changed. *Note: Configuration for SSL is available, and it has been tested successfully, but it has been commented out in the SecutiryConfig.java file for now.*