Delivery Documentation

Team JIC-8120

November 30, 2018

Team Members

Project Manager: Dennis Eddington
Backend Engineer: Erika Trejo
Backend Engineer: Joshua Gaul
Frontend Engineer: Mitchell Gant
Frontend Engineer: David Herszenhaut

Contents

Revision History

Version	Date	Name	Description
1	11/25/18	Dennis Eddington	Initial Document/Layout
2	11/27/18	Dennis Eddington	Update with Base Information
3	11/28/18	Dennis Eddington	Update Information
4	11/29/18	Dennis Eddington	Update Information
5	11/30/18	Dennis Eddington, Erika Trejo, Joshua Gaul, Mitchell Gant, David Herszenhaut	Finalized Documentation

1 Introduction

OpenEval is an application designed by Team JIC-8120 for our clients, the Office of Academic Effectiveness, at the Georgia Institute of Technology. This technology encompasses both an iOS and Web application that streamline the process of collecting midterm CIOS results from students. The purpose of this document is to detail setup parameters for future iterations of development with the product – OpenEval.

2 Release Notes

In this section of the documentation new features, bug fixes, as well as existing bug and defects will be outlines. Below is the original iteration plan for each phase of development for OpenEval.

Sprint	Weeks	Sprint Goal	User Stories / Features to be completed:
1	1-3	Class Registration	• As a professor, I want to be able to view all available GT courses so that I may register for a course. • As a professor, I want to search all available GT courses for my course so that I may register for my searched course. • As a professor, I want to be able to register a course so that I may begin to create course surveys for that course.
2	4-6	Professor account functionality including creating a course and creating surveys.	• As a user, I want to register an account so that I can publish a course survey and receive feedback. • As a professor, I want to create a survey, so that students in the course can fill it out and give feedback. • As a professor, I want to add a short response question to a survey I'm creating, so that students can give more detailed feedback about the course and instruction. • As a professor, I want to designate a

			time window for which the survey can be completed, so that I can control the amount of time students have to complete the survey.
3	7-9	Student account functionality including registering for courses and submitting surveys.	• As a student, I want to join my professor’s course, so that I can give that professor feedback. • As a student, I want to see the available surveys for all the courses I’ve joined, so that I can fill them out. • As a student, I want to see all the courses I’ve joined, so that I can then view its surveys. • As a student, I want to take a survey, so that I can provide feedback to my professor.
4	10-12	Add	• As a student, I want to submit a “Check-in”, so that I can give other students and the course difficulty, assignment difficulty, and course instruction at the current moment. • As a student, I want to view other students’ “Check-ins” so that I can gauge course and assignment difficulty. • As a professor, I want to edit a survey so that I can make changes to available class surveys. • As a professor, I want to set survey notifications to alert students of available surveys and their deadlines.
5	13-15	Testing, bug fixes, final check	• As a professor, I want the application to be intuitive so that I can easily receive course feedback. • As a student, I want the application to be intuitive so that I can easily offer course feedback.

The following will document the accomplishments of Sprint 5 as well as derivations taken in development from the original iteration plan.

Sprint 5 New Features:

- Implemented secure custom login protocol to authenticate multiple users synchronously
- Implemented notification system to blast email students in the class that a new survey has been posted

Sprint 5 Bug Fixes:

- Fixed an issue with iOS application that led to collecting the wrong class information for a given student
- Added API routes relating to hashed student information for secure delivery to and from network endpoints
- Added API routes for professors to obtain information regarding to the classes they were involved in
- Corrected algorithm for email notification that would occasionally send more than one email to the same student

Deprecated Features:

- “Check-in” feature
- Full control of “Edit Survey” feature
- Text Sentiment Analysis feature
- Delivery Specification/Formatting feature

3 Access to Server/iOS/Web Application Code

In this section of the documentation the proper way to access/install different elements of OpenEval will be explained.

Server:

All server code is hosted on github at the following repository - https://github.com/dedding4341/OpenEval

To access this code, ensure that you have the latest version of git installed to your machine. Following this, open command prompt or terminal and type the following – git version. The response should look like this (versioning may differ)…

```
git version 2.10.1.windows.1
```

After this create a new folder with any name then navigate to it using the following series of commands:

```
mkdir <insert file name here>
cd <insert file name here>
```

At this point we will now clone the project into this folder using the following command:

```
git clone https://github.com/dedding4341/OpenEval.git
```

The source code for the server application is now on your local machine. The server is currently deployed to a custom instance hosted via Salesforce Heroku. If you navigate to the Server folder you can replicate this deployment to any cloud service by creating an account and following the instructions provided.

iOS:

The features in the release are centered around two target users: professors and students. For professors, the application allows them register for a course, add a survey to the course that can then be seen by all of the students who are registered for the course and set a custom availability for when they want the survey to be administered. For students, the application can be used to view surveys released by the professor for a particular course and then submit a response to a survey based on its availability. There are currently no known bugs in the application but there are a lot of incomplete features that are not currently present in this release. On the professor side, these features include: reminder emails from professors to students, professors being able to view results of surveys, logging in as a professor, prepopulated registered courses, creating custom surveys and professors being able to export survey results. On the student side these features include: students being able to leave “check-ins” or small feedback forms that describe the difficulty of the course at a certain period of time at their own discretion, student login, student prepopulated registration of courses, and students being able to view data from previous check-ins.

iOS Install Guide:

Prerequisites: To be able to run the application in the simulator on your computer or download it to your device you must first own or have access to an Apple computer. You then must have XCode installed on your Apple computer. XCode is the IDE used to make iOS and Mac apps. If it is not downloaded, then it may take a while to install on your computer based on your internet speeds. Once these actions have been completed all that is left to do is to download the project from GitHub.

Dependent Libraries: There are no dependent third-party libraries that need to be installed for this portion of the application.

Download Instructions: The project can be download in its most up to date state at https://github.com/m-gant/OpenEvaliOSApplication

Build Instructions: Do not apply to this portion of the application.

Installation Instructions: Do not apply to this portion of the application.

Run Instructions: To build and run the application with a device simulator, you would first have to navigate to where the project was downloaded. Once you have done that, you would navigate to the .xcodeproj file and you could double-click that file and it would open the code for the application in XCode. Once you have the project opened you will see a play button in the top left corner of the window. You will also see a button with options for which device you would like to simulate the application on. Once you have chosen an appropriate device you can press the the play button and it will run the application in the simulator of your choosing.

Troubleshooting: If you come across any issues while attempting to build and run the application feel free to contact Mitchell Gant via email at mgant6@gatech.edu.

Web:

New software features for this release

- Finished the base version of the application:
- Rudimentary log in for students and professors
- Viewing a rudimentary list of all current courses
- View a user's currently registered courses
- View the active surveys for a course
- View the survey responses (for professors)
- Submit survey responses (for students)

Bug fixes made since the last release

- The search bar in all-classes.html now updates the class list automatically, without the user having to press a button to update the class list.
- The browser now displays a message when viewing a class that has no active surveys and when viewing a survey that has no responses.

Web Install Guide:

Prerequisites:

- A terminal
- A web browser

Dependent libraries:

- Node.js
- npm

Download instructions: The code for the web application can be found here or here. A copy of the code can be gotten by cloning the repository using git clone or by downloading a ZIP of the repository on GitHub.

Installation Instructions: In a terminal, go to the directory where the code has been downloaded and run npm install.

Run Instructions: Once the appropriate Node.js packages have been installed, start the application by running either node app.js or npm start. Open a browser and type http://localhost:3000/login in the address bar. From here, use one of the mock user accounts to navigate the application

- abray3, professor
- cperez3, student

Troubleshooting: If there is an error in starting the application, try running npm i npm to update npm and then running npm install to try installing the required packages again. Another common problem is pages not loading the correct information. For example, going from viewing one course's surveys to another course's surveys might still show the data from the previous course. If this happens. try refreshing the page.