

Software and System Test Plan

Employee Tracker

COP4331C Fall 2017

Team Name: Team 6

Team Members:

- Jordan Martin
- Brandon Bradley
- Ryan Hoeck
- Navon Francis
- Marc Simon
- David Simoneau

Contents of this Document

I. Test Plan

Overall Objective for Software Test Activity
Description of Test Environment
Overall Stopping Criteria
Description of Individual Test Cases
Appendices

II. Test Results

Description of the Actual Test Environment
Results of Individual Test Cases
Conclusion

I. Test Plan

Overall Objective for Software Test Activity

The software test effort will discover faults that could cause failures in the Employee Tracker mobile and web application. A successful test effort will be defined as one that finds faults that allows the development team to issue corrections leading to successful acceptance testing. The software test effort will be considered a failure if it finds few or no faults in the web and mobile application and fails acceptance testing. Manually testing will be done, however, this is to be noted that it will not be our main source of testing.

Through a set of Unit Tests we will aim to achieve small functionalities within components and ensure they work correctly. These functions are to be thoroughly tested so that basic actions (clicking buttons, pinning Firebase, etc.) work without a doubt. This will help us create more elaborate Integration Tests along the course of the test phase.

Through a set of Integration tests we will aim to achieve larger scale functionalities work, components are talking to each other, etc. We will comprise Integration tests around the bigger core functions of the application. Having previously mentioned thorough unit tests, this will better help us when comprising these tests.

Description of Test Environment

Jest, developed by Facebook and Enzyme, developed by AirBnB are popular unit testing software for React and Javascript components. Since React is developed by Facebook and Jest is included in React, using Jest to test our software is sensible. Facebook uses Jest to test their own React applications.

The hardware environment that the web application will be tested on will include Windows laptops, desktops, and Mac laptops and desktops. The development team is in possession of this hardware for the purposes of developing and testing the Employee Tracker web application.

- A. Windows
 - a. Laptop
 - i. Windows 8
 - ii. Windows 10
 - b. Desktop
 - i. Windows 10
- B. Mac
 - a. Laptop
 - i. macOS High Sierra
 - b. Desktop
 - i. macOS Sierra

As Windows includes Internet Explorer on the previous Windows 8 OS and Edge on the latest Windows 10 release, they will be included as native testing for the respective operating systems on the Employee Tracker web application. macOS includes the latest version of Safari on all recent releases of the OS and will be included as native testing on the Employee Tracker web application. This accounts for the last 5 years of Windows releases and the last 2 years of macOS releases. This means that we will be able to perform software testing on hardware in the native environment of the respective web browsers. We will be also be testing Chrome and Firefox in all environments. The latest stable version of each web browser we will be testing is Chrome (v.62.0.3202), Firefox (v.57.0), Microsoft Edge (v.41.16299.15), Safari (v.11.0.1), and Internet Explorer (v.11.0).

The hardware environment that the mobile application will be tested on will include Android and iOS devices. The development team is in possession of this hardware for the purposes of developing and testing the Employee Tracker mobile application.

- A. Android
 - a. Phone
 - i. Android KitKat (v.4.4.4)
 - ii. Android Nougat (v.7.1.2)
- B. iOS
 - a. Phone
 - i. iOS 9
 - ii. iOS 11

As both mobile operating systems are highly fragmented relative to the desktop operating system releases, the scope of the native mobile hardware testing will be less thorough. Fortunately there is a good spread of older and newer devices which will help ensure that performance testing can occur on older devices. This accounts for the last 4 years of mobile operating systems on Android and 2 years of iOS releases. This means we will be able to perform software testing on native hardware instead of the respective simulators.

The test team will consist of the developers on our team as outlined in previous deliverables. As we don't have the resources or manpower to form an independent test team, we will do the best we can with our available resources.

Stopping Criteria

When testing, we will determine when to stop when we have exhausted all options for that particular test case. We will NOT design test cases around our code just to make the tests work. We will create to the best of our ability fully in-depth test cases. When our tests fail, we will stop testing and find errors in code that we can easily fix. If this error is not easily fixable or the person testing is not knowledgeable in this area we will then bring it up in our stand up meetings. Then moving onto the next test, without having to fully stop and wait for someone to help.

As previously stated we will try to make very in-depth test cases, so we will most likely get at least one error. But, in the case that we do not, we will create a set of overlay tests on top of the

originals. We believe that testing is the most important part of making software work, hence very in-depth Unit and Integration Tests.

Many places in the workforce define “good enough to deliver” when you are getting a binary comparison when running a test in QA environment. Although, we do not have such an environment to test our software in, we will conduct full regression testing on features in our own production environments. We will carry out all necessary and unnecessary functions needed on each and component. Once we get our own sense of security that our version of this “binary comparison” test is sufficient, we will deem it good enough to deliver and let the customer decide.

Description of Individual Test Cases

Test ID	Test Objective	Test Description	Test Conditions	Expected Results
T.3.1.1	This test will determine if the login component of the mobile application correctly allows login and prevents unsuccessful logins.	This test will input strings of a valid username, password (employ,weak@\$\$), an invalid username, password (employ333, any\$#1s), and an valid username, invalid password (employ, wrong\$\$) into the login page of the mobile application. A uid will be returned on successful login.	Mobile Application only, application must be on login page.	Case 1: App receives uid, logs user in. Case 2: App does not receive uid, instead receives 401: Forbidden HTTP response, does not log user in. Case 3: App does not receive uid, instead receives 401: Forbidden HTTP response, does not log user in.
T.3.1.1.2	This test will determine if the mobile application correctly sends an updated GPS location periodically	After login and tracking have begun on a test phone, actual gps location of phone sent back to the database will be compared for accuracy and compliance. A GPS location will be returned periodically for a successful test run.	Mobile Application, Firebase Database	Database receives location data after user login at a regular interval.
T.3.1.2	This test will determine if the web	This is a mix of a observational	Web application,	The unit test

	application correctly shows a list of employees, as well as their locations on a map	and unit test. For the list, the test will pull the list of employees from the database, and compare it directly with the list of employees in the web application to see if they match. The test for the map being populated will be an observational test, where us (the developers) will ensure that the map is displaying the location of the employees correctly by comparing the location in the database to the location on the map.	after build before deployment to firebase.	returns a "Pass" and the employees coordinates match with the coordinates on the map.
T.3.1.3	This test will determine if the applications correctly accept valid passwords that conform to a standard, and deny invalid/nonconforming ones.	This test will input strings of a valid username, password (employ,weak@\$1\$), an invalid username, password (employ333, any\$#1s), and an valid username, invalid password (employ, wrongpa\$\$) into the login page of the mobile application. A uid will be returned on successful login. The standard is the password must have a lowercase letter, a number, and a symbol	Mobile and web application.	The applications accept passwords that conform to the rules, otherwise they deny them.
T.3.1.4	This test will determine if the mobile application will continue to collect gps coordinates for a maximum of 30 minutes on data/connection loss.	The test will begin after login and tracking has begun on a test phone. The connection to the database will be cut off and the phone will continue storing location data in a local cache. A timer triggered by the loss of connection, will begin and once the maximum time limit of 30 minutes has been reached, the user will be logged out.	Mobile Application, Firebase Database	After disconnection from the database the test phone local cache will begin locally storing location data. User will be logged out on Mobile application after greater than 30 minutes of disconnection from database.

T.3.1.5	This test will determine the success of the alert system in the applications. If the mobile application fails to connect for 30 minutes, camera, or gps fails, an alert will be generated.	This test will disable data transfer to the firebase database. It will then start a timer at the time of disconnect. If there is no connection within 30 minutes, the employee will be forcibly logged out. This test will also trigger a controlled camera failure and gps failure. The test will ensure that a text notification of "Connection/camera/gps failure" to the employee. If a connection is present, the message also get sent to the database.	This test will be run in the mobile application, both emulator and a physical device before building.	A error message "Connection/camera/gps failure" is generated, and if a connection is present, then the userMessage field of the respective employee's database in the server now contains that same message.
T.3.1.6	This test will determine if the mobile application correctly sends GPS locations collected when disconnected from the internet, upon successful reconnection.	The test will begin after login and tracking has begun on a test phone. The connection to the database will be cut off and the phone will continue storing location data in a local cache. After a time less than 30 minutes but greater than two location update periods, connection will be reestablished with the database and locally cached location data will be sent to the database along with new updated positions.	Mobile Application, Firebase Database	After disconnection from the database the test phone local cache will begin locally storing location data. Upon reconnection, phone sends locally cached location data, along with new updated location data periodically.
T.3.2.1	This test will determine if the mobile application correctly passes inputted credentials to firebase, and firebase then correctly verifies the employee.	This test will input strings of a valid username, password (employ,weak@\$1\$), an invalid username, password (employ333, any\$#1s), and an valid username, invalid password (employ, wrongpa\$\$) into the login page of the mobile application. A uid will	Mobile and Web application, and Firebase for authentication	Case 1: App receives uid, logs user in. Case 2: App does not receive uid, instead receives 401: Forbidden HTTP

		be returned on successful login.		response, does not log user in. Case 3: App does not receive uid, instead receives 401: Forbidden HTTP response, does not log user in.
T.3.2.2	This test will determine if the mobile application correctly interfaces with the firebase database through RESTful requests.	This test will read all connections to the database for a period of 10 minutes. It will then attempt to parse them as REST requests.	The mobile application, after building	The connections are valid REST requests. The test then will return a "Pass" message.
T.3.2.3	This test determines that the precision of the GPS locations is 6 decimal places, and that it gets sent to the firebase database periodically.	After login and tracking has begun on a test phone, actual gps location of phone sent back to the database will be compared for accuracy and precision of 6 decimal places. A GPS location will be returned periodically for a successful test run.	Mobile Application, Firebase Database	Database receives 6-digit decimal precision location data after user login at a regular interval and is returned.
T.3.2.4	This test will determine if the web application correctly interfaces with the firebase database through RESTful requests.	This test will read all connections to the database for a period of 10 minutes. It will then attempt to parse them as REST requests.	The mobile application, after building	The connections are valid REST requests. The test then will return a "Pass" message.
T.3.4.1	This test will determine if there are three groups available to be logged in.	This test is purely observational from the developer view. If there are three classes that represent Employer, Employee, Admin, then the test passes.	The Mobile, Web application, and the Firebase database	There are three individual groups/classes implemented.
T.3.4.3	This test will verify that Employees do not have access to Employer permissions and vice versa.	For this test, a logged in Employee should not have access to "add/remove employee" that the Employer has. A POST request containing the method	A employee is logged in via the mobile application	A 401: Forbidden HTTP response.

		updateEmployee() will be sent from the mobile application. If it returns anything but an 401 Forbidden error, the test will fail.		
T.3.4.4	This test will verify that a training manual is provided.	This is another observation test. At the time of client implementation, a training manual is provided. If the manual is not provided, the test fails.	The client chooses to implement the software system	A training manual is provided
T.3.5.1	This test will determine if the training manual has been updated when the software updates.	This test compares the commit history between the code and the manual. If no commits to the manual occur within 10 code commits, the test fails.	The developer space/repository system is in use	The commits to the manual should occur within every 10 code commits. If not, the test fails.
T.3.5.2	This test will verify that the documentation has images and diagrams.	This test will grep the documentation file for any image extensions. It will also verify that those referenced image files exist in the referenced directory	This test will be run when we build the applications.	Image extensions with existing referenced files will exist, resulting in a "Pass" from the test.
T.3.5.3	This test will verify that the documentation is concise.	This test will compare the word count of the documentation to the number 15000. This number comes from the average of 300-350 words per page, and the maximum page length we set as 50 pages.	During after build time of mobile and web applications.	The test returns "Pass" if the word count is less than 15000.
T.3.5.4	This test will verify that the documentation is presented in layman's terms, no jargon in it.	This is an observational test. This test involves us (the developers) reading through the documentation and ensuring that there is no complex technical terms.	After final build of applications, before publishing work.	There are no technical terms involved in the documentation.
T.3.5.5	This test will verify that all interactable portions of the software are documented with explanations.	Another observational test. This involves us (the developers) going through the documentation and ensuring that every time we have a updated render() in the code,	After final build of applications, before publishing work.	There is a corresponding description of every render() update in the

		there is a corresponding description in the documentation.		code to a paragraph in the documentation.
T.3.6.1	This test will determine if degree of GPS precision is correctly 6 decimal places for latitude and longitude.	After login and tracking has begun on a test phone, actual gps location of phone sent back to the database will be compared for accuracy and precision of 6 decimal places. Location data will be compared on phone and from database to actual latitude and longitude coordinates. Matching longitude and latitude from phone, Map View, and database will provide a successful test run.	Mobile Application, Web Application, Firebase Database	6 digit longitude and latitude location data sent from phone will match 6 digit longitude and latitude data on database and Map view. This will be in format (28.6024 N, 81.2001 W)
T.3.6.2	This test will determine if GPS coordinates in the form of latitude and longitude for each individual employee are successfully retained in the database.	User will login to test phone and begin GPS tracking. A successful run will consist of correct data being logged in database.	Mobile Application, Firebase Database	Location data from mobile phone will be correctly logged in Firebase database.
T.3.6.3	This test will determine if the employee's name and unique ID are retained together in the firebase database.	Using our own dummy data of the six of us will allow us to automate this and quickly discover whether or not the test cases are off. This test will iterate through all location entries in the database, and ensure that each employeeName had a corresponding unique ID.	Firebase database test with string comparisons of our own information.	Each employee has a unique ID assigned to them in the database. A result of "Pass" is generated on success.
T.3.6.4	This test will determine if the date and time of the gathered GPS coordinates are retained alongside the rest of the gathered data in the database.	This test will iterate through all location entries in the database, and ensure that each location entry had a corresponding timestamp in ISO format	Database connection, when database is filled with dummy data.	Each location has a corresponding timestamp. A result of "Pass" if this is true. Otherwise, "Fail"

T.3.6.5	This test will determine if all logins are successfully logged in the database.	Multiple correct and incorrect logins will be conducted on the Web and Mobile applications, and data will be sent to the database. Database will then be checked against login attempts for corresponding data.	Mobile, Web application, Firebase for data logging	Case 1: User logs in, data is logged correctly in database. Case 2: User attempts to log in using incorrect information, database logs incorrect attempt with corresponding data.
T.3.7.1	This test will determine that the database will provide long lasting storage for the information gathered.	Documentation will be reviewed to show that our version of our database is stable and provides longevity.	Periodic installs of our backups on test servers to ensure backups are working correctly	Test server with backup successfully passes all automated testing.
T.3.7.2	This test will verify that a system administrator has access to the firebase database and can debug.	Observational Test. One or more intentional bugs to ensure proper response.	Mobile Application, Web Application, Firebase Database	All intentional bugs (viewed only by developers) are viewable.
T.3.7.3	This test will verify that the Employer utilizing this software will provide the employees with all necessary technology.	Peer Review System. This will be a user response test in which we provide a way to give feedback to us to ensure the safety and privacy of our users.	Mobile Application, Web Application	Periodic automated tests on the feedback system to ensure developers are receiving feedback by its users.
T.3.8.1	This test will determine if access to the system is successfully controlled and secured by logins for each client.	Ensures only the correct user is viewing information they are allowed to view and nothing else. The test will ensure that the login for a employee only retrieves data for that employee, and no one else. This will be done by comparing the information returned on login to the login	Mobile Application, Web Application, Firebase Database	Case 1: URL testing of what might allow a view of another user. Case 2: Validating users in the database are

		status of the employee in the database.		not connected to other users' information.
T.3.8.2	This test will determine if the client's data is successfully isolated from other clients through logins.	We are ensuring the user can only log in to their own accounts.	Mobile Application, Web Application, Firebase Database	Same as T.3.8.1
T.3.8.3	This test will verify that the client's data is encrypted with the AES 256-bit standard.	Implementation can be seen within the firebase data. Observation.	Firebase Database	We will have a dummy row of a user in the database and will string check known values with the database to ensure none match, if so, then the encryption is not occurring.
T.3.8.4	This test will determine if the database is hardened against SQL injection attacks.	A couple standard attempts in user input fields to ensure that the sql fails to be injected.	Mobile Application, Web Application	There are many testing suites to test this, which will be implemented, which could include oWasp or sqlmap
T.3.8.5	This test will determine if the system is successfully backed up, and is backed up periodically.	During start up of the system, we will make a backup and rerun the application to ensure that the backups are in fact working and then periodically check on whether firebase is still making backups.	Firebase Database, Mobile and Web Application	All automated and observational testing works on the live backup environment.
T.3.8.6	This test will verify that the employee using the web application is unable to spoof their location.	We will test current applications that allow gps spoofing and check whether the system is aware of the spoofing. The test will also tie into Android's SafetyNet and Apple's equivalent security system. If the test detects root or jailbreak privileges, or that	Mobile Application. Firebase Database, after building before deploying.	Our testing will be more hands on, using sqlmap as well as other applications to spoof our own location and determine

		developer settings are enabled and thus location spoofing, the app will generate an alert message and send it to the database. There is also observational testing, where the Employer ensures to provide the employees with locked down devices.		whether or not the system is aware.
T.3.9.1	This test will determine if the Employee Tracker software is available as a web and mobile application.	Observation and Administration testing will test this.	Mobile and Web Application. Firebase Database,	All automated testing and observational testing will must be passed to ensure both applications are up and running.
T.3.9.2	This test will verify that the software is maintained by the people in Group 6: Employee Tracker, in class COP-4331 during the Fall 2017 Semester.	Observation. GitHub allows users to see who is editing and when it has occurred.	Github, git log.	Observation will show whether or not updates have been made as and the whole team is developing the product.
T.3.9.3	This test will verify that the testing suite is automated.	Other than observational testing, we will be able to run scripts that run automatically and print in a debug log of any errors that could have occurred.	Debug logs within our applications	This will be observational. We will check the debug logs for any detrimental activity and react accordingly.

Trace of Individual Test Cases to the Requirements Traceability Matrix

Req ID	Requirement Description	Architecture Reference	Design Reference (component/module)	Test Case Reference	Status
3.1.1	The Employee Tracker shall allow the Employee to log in, update location, and see other users using the mobile application	Employee Login, Tracking status	Login <<interface>>, Mobile Application	T.3.1.1	
3.1.2	The Employee Tracker shall allow the Manager to see updates from the employee on the web application.	Employer Login, Select Any Employee	View, Web Application	T.3.1.2	
3.1.3	The login username and password will be verified to be valid according to the requirements set by the company.	Employee Login, Customer Login	Connection, Firebase Database	T.3.1.3	
3.1.4	The mobile application will continue running and collecting location coordinates for up to 30 minutes if loss of internet connectivity occurs. After 30 minutes, the system alerts both the user and the employer that the device has not had a connection for 30+ minutes.	Checks Tracking status, Employees Have Location turned on.	Connection, Mobile Application	T.3.1.4	
3.1.5	The mobile application will notify the user and the employer if the mobile device's GPS or camera malfunctions.	Clicks Employee Details, Turns Tracking on/off, Makes changes to the database information.	Web application, Mobile application	T.3.1.5	
3.1.6	The mobile application will send the locally stored locations to the server upon successful reconnection.	Makes changes to the Database Information.	Mobile Application	T.3.1.6	
3.2.1	The Employee Tracker application shall check username and password for each user login. The username and password inputs will be sent as strings to be checked and be validated with Firebase. The data must be exact as it is credentials for a user.	Employee Logs in, Customer Logs in	Connection, Firebase Database	T.3.2.1	

3.2.2	The Employee Tracker application shall fetch, the (fetch method), and post requests to and from Firebase to validate against the frontend that what the user input has integrity.	Retrieves Database Information, Makes changes to Database Information,	Connection	T.3.2.2	
3.2.3	Upon successful login, the mobile application shall send its location with 6 points of precision to the server immediately, and every 10 minutes thereafter, for the duration the user is logged in, or until a period of 30 minutes of no connectivity to the server passes.	Check Tracking Status, Makes Changes to database Information	Mobile Applications Location	T.3.2.3	
3.2.4	All data transfers between the Firebase server and the web or mobile application shall be through a RESTful API.	Database, Retrieves Database Information.	Connection	T.3.2.4	
3.4.1	The Employee Tracker Application shall have these 3 user groups: The Employer, Employee, and Admin.	Employee, Customer, Admin	Web application, Mobile Application, Firebase Database	T.3.4.1	
3.4.3	The Employee Tracking App shall prevent misuse by the user groups	Turns tracking on off, makes changes to database information.	Connection, Login, Mobile application	T.3.4.3	
3.4.4	Users shall be provided training and copies of the Employee Tracking Application manual, thus minimal knowledge of the system is required. The only ability required is being able to follow a procedure outlined in the manual, and being able to read at smallest an 11pt font.	Database, Client-Server	Documentation	T.3.4.4	
3.5.1	The documentation shall be stored with the software and updated in accordance to functional changes of the software.	Documentation	Documentation	T.3.5.1	
3.5.2	The documentation will have pictures and diagrams of the software, with footnotes explaining what each button does.	Documentation	Documentation	T.3.5.2	
3.5.3	The documentation should be concise and not span hundreds of pages.	Documentation	Documentation	T.3.5.3	
3.5.4	The documentation shall have no usage of legalese or other complicated technical terms or jargon that would confuse a user.	Documentation	Documentation	T.3.5.4	
3.5.5	The documentation should explain all parts of the software the user can interact with.	Documentation	Documentation	T.3.5.5	

3.6.1	The degree of GPS precision shall be 6 decimal places for latitude and longitude. For example, (28.6024 N, 81.2001 W) which are the GPS coordinates for the University of Central Florida. The accuracy of the GPS data will vary depending location, weather and the number of satellites in line of sight to the GPS hardware on the mobile device. The Google Maps API will be used to facilitate this requirement.	Database, Retrieves Database Information	Documentation	T.3.6.1	
3.6.2	GPS coordinates in the form of latitude and longitude for each individual employee must be retained in the database.	Database, Retrieves Database Information	Firebase Database, Algorithm	T.3.6.2	
3.6.3	The employee's name and unique ID will need to be retained together in the database.	Database, Make changes to the database.	Database	T.3.6.3	
3.6.4	The date and time of the gathered GPS coordinates shall be retained alongside the rest of the gathered data in the database.	Database	Database, User,	T.3.6.4	
3.6.5	Logins shall be retained in the database.	Client-Server, Repository	Database, Login	T.3.6.5	
3.7.1	A database shall be required for the backend to provide long term storage for our client's data. As our client's grow in number the costs associated with the Firebase database platform we chose will grow as well.	Database	Firebase Database	T.3.7.1	
3.7.2	A database administrator(s) will be needed to perform maintenance on the system as required. The system may need to be debugged at times and this action will require skilled personnel.	Database	Firebase Database	T.3.7.2	
3.7.3	The Employee Tracker application shall require the client to have access to computers and the client's employees to have access to mobile devices. We will not need to provide any hardware to the clients for the application to function as intended. In addition, we will not require physical space or supporting amenities as the client's business will cover those aspects of their operation.	Subscriber	Web Application, Database	T.3.7.3	

3.8.1	Access to the system shall be controlled and secured by logins for each client.	Server-Client	Firebase Database	T.3.8.1	
3.8.2	Client's data shall be isolated from other clients through logins.	Server-Client	Firebase Database	T.3.8.2	
3.8.3	Client's data shall be encrypted with 256-bit AES standard	Database	Firebase Database	T.3.8.3	
3.8.4	Database shall be tested against SQL injection	Database	Firebase Database	T.3.8.4	
3.8.5	Automatic backups shall be performed once per day offsite (Firebase cost dependent). Backups are off site on Google's servers and protected from water, fire and other natural disasters. Data can be recovered if needed at any time.	Database	Firebase Database	T.3.8.5	
3.8.6	Employee's shall not spoof their location.	Server-Client	Mobile Application	T.3.8.6	
3.9.1	The Employee Tracker Application shall be available as a web application and mobile application.	Client-Server, Repository	Web Application, Mobile Application	T.3.9.1	
3.9.2	Software shall be maintained by us, the software creators.	Repository		T.3.9.2	
3.9.3	Automation of bug-detection will be in place	Client-Server, Repository, Database	Firebase Database, UI, Web/Mobile Applications	T.3.9.3	
3.9.4	Downtime must be minimal (less than 30 minutes a week during working hours), Response time to issues should be within the hour of the issue arising.	Database	Connection, Firebase Database	T.3.9.4	

II. Test Results

Description of the Actual Test Environment

Thus far, the planned testing has occurred exactly as expected for our current system. However, this is mostly due to the fact that our testing automation has not been completed as of yet. We are not currently following the Test-Driven Development model and are mostly focused on module integration.

Results of Individual Test Cases

Most of the test cases have yet to be completed as we are not currently at a stage where testing is a priority.

Test ID	Test Objective	Test Description	Test Conditions	Expected Results	Actual Results
T.3.1.1	This test will determine if the login component of the mobile application correctly allows login and prevents unsuccessful logins.	This test will input strings of a valid username, password (employ,weak@\$\$), an invalid username, password (employ333, any\$#1s), and an valid username, invalid password (employ, wrong\$\$) into the login page of the mobile application. A uid will be returned on successful login.	Mobile Application only, application must be on login page.	Case 1: App receives uid, logs user in. Case 2: App does not receive uid, instead receives 401: Forbidden HTTP response, does not log user in. Case 3: App does not receive uid, instead receives 401: Forbidden HTTP response, does not log user in.	N/A
T.3.1.1.2	This test will determine if the mobile application correctly sends an updated GPS location periodically	After login and tracking have begun on a test phone, actual gps location of phone sent back to the database will be compared for accuracy and compliance. A GPS location will be returned periodically for a successful test run.	Mobile Application, Firebase Database	Database receives location data after user login at a regular interval.	N/A
T.3.1.2	This test will determine if the web application correctly shows a list of employees, as well as their locations on a map	This is a mix of a observational and unit test. For the list, the test will pull the list of employees from the database, and compare it directly with the list of employees in the web application to see if they match. The test for the map being populated will be an observational test, where us (the developers) will ensure that the map is displaying the location of	Web application, after build before deployment to firebase.	The unit test returns a "Pass" and the employees coordinates match with the coordinates on the map.	N/A

		the employees correctly by comparing the location in the database to the location on the map.			
T.3.1.3	This test will determine if the applications correctly accept valid passwords that conform to a standard, and deny invalid/nonconforming ones.	This test will input strings of a valid username, password (employ,weak@\$1\$), an invalid username, password (employ333, any\$#1s), and an valid username, invalid password (employ, wrongpa\$\$) into the login page of the mobile application. A uid will be returned on successful login. The standard is the password must have a lowercase letter, a number, and a symbol	Mobile and web application.	The applications accept passwords that conform to the rules, otherwise they deny them.	N/A
T.3.1.4	This test will determine if the mobile application will continue to collect gps coordinates for a maximum of 30 minutes on data/connection loss.	The test will begin after login and tracking has begun on a test phone. The connection to the database will be cut off and the phone will continue storing location data in a local cache. A timer triggered by the loss of connection, will begin and once the maximum time limit of 30 minutes has been reached, the user will be logged out.	Mobile Application, Firebase Database	After disconnection from the database the test phone local cache will begin locally storing location data. User will be logged out on Mobile application after greater than 30 minutes of disconnection from database.	N/A

T.3.1.5	This test will determine the success of the alert system in the applications. If the mobile application fails to connect for 30 minutes, camera, or gps fails, an alert will be generated.	This test will disable data transfer to the firebase database. It will then start a timer at the time of disconnect. If there is no connection within 30 minutes, the employee will be forcibly logged out. This test will also trigger a controlled camera failure and gps failure. The test will ensure that a text notification of "Connection/camera/gps failure" to the employee. If a connection is present, the message also get sent to the database.	This test will be run in the mobile application, both emulator and a physical device before building.	A error message "Connection/camera/gps failure" is generated, and if a connection is present, then the userMessage field of the respective employee's database in the server now contains that same message.	N/A
T.3.1.6	This test will determine if the mobile application correctly sends GPS locations collected when disconnected from the internet, upon successful reconnection.	The test will begin after login and tracking has begun on a test phone. The connection to the database will be cut off and the phone will continue storing location data in a local cache. After a time less than 30 minutes but greater than two location update periods, connection will be reestablished with the database and locally cached location data will be sent to the database along with new updated positions.	Mobile Application, Firebase Database	After disconnection from the database the test phone local cache will begin locally storing location data. Upon reconnection, phone sends locally cached location data, along with new updated location data periodically.	N/A
T.3.2.1	This test will determine if the mobile application correctly passes inputted credentials to firebase, and firebase then correctly verifies the employee.	This test will input strings of a valid username, password (employ,weak@\$1\$), an invalid username, password (employ333, any\$#1s), and an valid username, invalid password (employ, wrongpa\$\$) into the login	Mobile and Web application, and Firebase for authentication	Case 1: App receives uid, logs user in. Case 2: App does not receive uid, instead receives 401: Forbidden HTTP response, does not log user in. Case 3: App does not receive uid, instead	N/A

		page of the mobile application. A uid will be returned on successful login.		receives 401: Forbidden HTTP response, does not log user in.	
T.3.2.2	This test will determine if the mobile application correctly interfaces with the firebase database through RESTful requests.	This test will read all connections to the database for a period of 10 minutes. It will then attempt to parse them as REST requests.	The mobile application, after building	The connections are valid REST requests. The test then will return a "Pass" message.	N/A
T.3.2.3	This test determines that the precision of the GPS locations is 6 decimal places, and that it gets sent to the firebase database periodically.	After login and tracking has begun on a test phone, actual gps location of phone sent back to the database will be compared for accuracy and precision of 6 decimal places. A GPS location will be returned periodically for a successful test run.	Mobile Application, Firebase Database	Database receives 6-digit decimal precision location data after user login at a regular interval and is returned.	N/A
T.3.2.4	This test will determine if the web application correctly interfaces with the firebase database through RESTful requests.	This test will read all connections to the database for a period of 10 minutes. It will then attempt to parse them as REST requests.	The mobile application, after building	The connections are valid REST requests. The test then will return a "Pass" message.	N/A
T.3.4.1	This test will determine if there are three groups available to be logged in.	This test is purely observational from the developer view. If there are three classes that represent Employer, Employee, Admin, then the test passes.	The Mobile, Web application, and the Firebase database	There are three individual groups/classes implemented.	N/A
T.3.4.3	This test will verify that Employees do not have access to Employer permissions and vice versa.	For this test, a logged in Employee should not have access to "add/remove employee" that the Employer has. A POST request containing the method updateEmployee() will be sent from the mobile	A employee is logged in via the mobile application	A 401: Forbidden HTTP response.	N/A

		application. If it returns anything but an 401 Forbidden error, the test will fail.			
T.3.4.4	This test will verify that a training manual is provided.	This is another observation test. At the time of client implementation, a training manual is provided. If the manual is not provided, the test fails.	The client chooses to implement the software system	A training manual is provided	N/A
T.3.5.1	This test will determine if the training manual has been updated when the software updates.	This test compares the commit history between the code and the manual. If no commits to the manual occur within 10 code commits, the test fails.	The developer space/repository system is in use	The commits to the manual should occur within every 10 code commits. If not, the test fails.	N/A
T.3.5.2	This test will verify that the documentation has images and diagrams.	This test will grep the documentation file for any image extensions. It will also verify that those referenced image files exist in the referenced directory	This test will be run when we build the applications.	Image extensions with existing referenced files will exist, resulting in a "Pass" from the test.	N/A
T.3.5.3	This test will verify that the documentation is concise.	This test will compare the word count of the documentation to the number 15000. This number comes from the average of 300-350 words per page, and the maximum page length we set as 50 pages.	During after build time of mobile and web applications.	The test returns "Pass" if the word count is less than 15000.	N/A
T.3.5.4	This test will verify that the documentation is presented in layman's terms, no jargon in it.	This is an observational test. This test involves us (the developers) reading through the documentation and ensuring that there is no complex technical terms.	After final build of applications, before publishing work.	There are no technical terms involved in the documentation.	N/A

T.3.5.5	This test will verify that all interactable portions of the software are documented with explanations.	Another observational test. This involves us (the developers) going through the documentation and ensuring that every time we have a updated render() in the code, there is a corresponding description in the documentation.	After final build of applications, before publishing work.	There is a corresponding description of every render() update in the code to a paragraph in the documentation.	N/A
T.3.6.1	This test will determine if degree of GPS precision is correctly 6 decimal places for latitude and longitude.	After login and tracking has begun on a test phone, actual gps location of phone sent back to the database will be compared for accuracy and precision of 6 decimal places. Location data will be compared on phone and from database to actual latitude and longitude coordinates. Matching longitude and latitude from phone, Map View, and database will provide a successful test run.	Mobile Application, Web Application, Firebase Database	6 digit longitude and latitude location data sent from phone will match 6 digit longitude and latitude data on database and Map view. This will be in format (28.6024 N, 81.2001 W)	N/A
T.3.6.2	This test will determine if GPS coordinates in the form of latitude and longitude for each individual employee are successfully retained in the database.	User will login to test phone and begin GPS tracking. A successful run will consist of correct data being logged in database.	Mobile Application, Firebase Database	Location data from mobile phone will be correctly logged in Firebase database.	N/A
T.3.6.3	This test will determine if the employee's name and unique ID are retained together in the firebase database.	Using our own dummy data of the six of us will allow us to automate this and quickly discover whether or not the test cases are off. This test will iterate through all location entries in the database, and ensure that	Firebase database test with string comparisons of our own information.	Each employee has a unique ID assigned to them in the database. A result of "Pass" is generated on success.	N/A

		each employeeName had a corresponding unique ID.			
T.3.6.4	This test will determine if the date and time of the gathered GPS coordinates are retained alongside the rest of the gathered data in the database.	This test will iterate through all location entries in the database, and ensure that each location entry had a corresponding timestamp in ISO format	Database connection, when database is filled with dummy data.	Each location has a corresponding timestamp. A result of "Pass" if this is true. Otherwise, "Fail"	N/A
T.3.6.5	This test will determine if all logins are successfully logged in the database.	Multiple correct and incorrect logins will be conducted on the Web and Mobile applications, and data will be sent to the database. Database will then be checked against login attempts for corresponding data.	Mobile, Web application, Firebase for data logging	Case 1: User logs in, data is logged correctly in database. Case 2: User attempts to log in using incorrect information, database logs incorrect attempt with corresponding data.	N/A
T.3.7.1	This test will determine that the database will provide long lasting storage for the information gathered.	Documentation will be reviewed to show that our version of our database is stable and provides longevity.	Periodic installs of our backups on test servers to ensure backups are working correctly	Test server with backup successfully passes all automated testing.	N/A
T.3.7.2	This test will verify that a system administrator has access to the firebase database and can debug.	Observational Test. One or more intentional bugs to ensure proper response.	Mobile Application, Web Application, Firebase Database	All intentional bugs (viewed only by developers) are viewable.	N/A
T.3.7.3	This test will verify that the Employer utilizing this software will provide the	Peer Review System. This will be a user response test in which we provide a way to give feedback to us to ensure the safety and privacy of our	Mobile Application, Web Application	Periodic automated tests on the feedback system to ensure developers are receiving feedback by	N/A

	employees with all necessary technology.	users.		its users.	
T.3.8.1	This test will determine if access to the system is successfully controlled and secured by logins for each client.	Ensures only the correct user is viewing information they are allowed to view and nothing else. The test will ensure that the login for a employee only retrieves data for that employee, and no one else. This will be done by comparing the information returned on login to the login status of the employee in the database.	Mobile Application, Web Application, Firebase Database	Case 1: URL testing of what might allow a view of another user. Case 2: Validating users in the database are not connected to other users' information.	N/A
T.3.8.2	This test will determine if the client's data is successfully isolated from other clients through logins.	We are ensuring the user can only log in to their own accounts.	Mobile Application, Web Application, Firebase Database	Same as T.3.8.1	N/A
T.3.8.3	This test will verify that the client's data is encrypted with the AES 256-bit standard.	Implementation can be seen within the firebase data. Observation.	Firebase Database	We will have a dummy row of a user in the database and will string check known values with the database to ensure none match, if so, then the encryption is not occurring.	N/A
T.3.8.4	This test will determine if the database is hardened against SQL injection attacks.	A couple standard attempts in user input fields to ensure that the sql fails to be injected.	Mobile Application, Web Application	There are many testing suites to test this, which will be implemented, which could include oWasp or sqlmap	N/A

T.3.8.5	This test will determine if the system is successfully backed up, and is backed up periodically.	During start up of the system, we will make a backup and rerun the application to ensure that the backups are in fact working and then periodically check on whether firebase is still making backups.	Firebase Database, Mobile and Web Application	All automated and observational testing works on the live backup environment.	N/A
T.3.8.6	This test will verify that the employee using the web application is unable to spoof their location.	We will test current applications that allow gps spoofing and check whether the system is aware of the spoofing. The test will also tie into Android's SafetyNet and Apple's equivalent security system. If the test detects root or jailbreak privileges, or that developer settings are enabled and thus location spoofing, the app will generate an alert message and send it to the database. There is also observational testing, where the Employer ensures to provide the employees with locked down devices.	Mobile Application. Firebase Database, after building before deploying.	Our testing will be more hands on, using sqlmap as well as other applications to spoof our own location and determine whether or not the system is aware.	N/A
T.3.9.1	This test will determine if the Employee Tracker software is available as a web and mobile application.	Observation and Administration testing will test this.	Mobile and Web Application. Firebase Database,	All automated testing and observational testing will must be passed to ensure both applications are up and running.	N/A
T.3.9.2	This test will verify that the software is maintained by the people in Group 6: Employee Tracker,	Observation. GitHub allows users to see who is editing and when it has occurred.	Github, git log.	Observation will show whether or not updates have been made as and the whole team is	Ryan Hoeck, 2017-11-20, Pass.

	in class COP-4331 during the Fall 2017 Semester.			developing the product.	
T.3.9.3	This test will verify that the testing suite is automated.	Other than observational testing, we will be able to run scripts that run automatically and print in a debug log of any errors that could have occurred.	Debug logs within our applications	This will be observational. We will check the debug logs for any detrimental activity and react accordingly.	N/A
T.3.9.4	This test will determine if the downtime and response time is within the acceptable range.	Peer Review System. Half of our team can startup a development system and ensure that the response is valid and acceptable according to the specifications.	Backup system of the whole application and database.	Bringing down and bringing back up the whole system and a full run of the test suite to acquire data for the efficiency and the downtime during the process.	N/A

Req ID	Requirement Description	Architecture Reference	Design Reference (component/module)	Test Case Reference	Status
3.1.1	The Employee Tracker shall allow the Employee to log in, update location, and see other users using the mobile application	Employee Login, Tracking status	Login <<interface>>, Mobile Application	T.3.1.1	F
3.1.2	The Employee Tracker shall allow the Manager to see updates from the employee on the web application.	Employer Login, Select Any Employee	View, Web Application	T.3.1.2	F
3.1.3	The login username and password will be verified to be valid according to the requirements set by the company.	Employee Login, Customer Login	Connection, Firebase Database	T.3.1.3	F
3.1.4	The mobile application will continue running and collecting location coordinates for up to 30 minutes if loss of internet connectivity occurs. After 30 minutes, the system alerts both the user and the employer that the device has not had a connection for 30+	Checks Tracking status, Employees Have Location turned on.	Connection, Mobile Application	T.3.1.4	F

	minutes.				
3.1.5	The mobile application will notify the user and the employer if the mobile device's GPS or camera malfunctions.	Clicks Employee Details, Turns Tracking on/off, Makes changes to the database information.	Web application, Mobile application	T.3.1.5	F
3.1.6	The mobile application will send the locally stored locations to the server upon successful reconnection.	Makes changes to the Database Information.	Mobile Application	T.3.1.6	F
3.2.1	The Employee Tracker application shall check username and password for each user login. The username and password inputs will be sent as strings to be checked and be validated with Firebase. The data must be exact as it is credentials for a user.	Employee Logs in, Customer Logs in	Connection, Firebase Database	T.3.2.1	F
3.2.2	The Employee Tracker application shall fetch, the (fetch method), and post requests to and from Firebase to validate against the frontend that what the user input has integrity.	Retrieves Database Information, Makes changes to Database Information,	Connection	T.3.2.2	P
3.2.3	Upon successful login, the mobile application shall send its location with 6 points of precision to the server immediately, and every 10 minutes thereafter, for the duration the user is logged in, or until a period of 30 minutes of no connectivity to the server passes.	Check Tracking Status, Makes Changes to database Information	Mobile Applications Location	T.3.2.3	F
3.2.4	All data transfers between the Firebase server and the web or mobile application shall be through a RESTful API.	Database, Retrieves Database Information.	Connection	T.3.2.4	P
3.4.1	The Employee Tracker Application shall have these 3 user groups: The Employer, Employee, and Admin.	Employee, Customer, Admin	Web application, Mobile Application, Firebase Database	T.3.4.1	P
3.4.3	The Employee Tracking App shall prevent misuse by the user groups	Turns tracking on off, makes changes to database information.	Connection, Login, Mobile application	T.3.4.3	F

3.4.4	Users shall be provided training and copies of the Employee Tracking Application manual, thus minimal knowledge of the system is required. The only ability required is being able to follow a procedure outlined in the manual, and being able to read at smallest an 11pt font.	Database, Client-Server	Documentation	T.3.4.4	P
3.5.1	The documentation shall be stored with the software and updated in accordance to functional changes of the software.	Documentation	Documentation	T.3.5.1	P
3.5.2	The documentation will have pictures and diagrams of the software, with footnotes explaining what each button does.	Documentation	Documentation	T.3.5.2	F
3.5.3	The documentation should be concise and not span hundreds of pages.	Documentation	Documentation	T.3.5.3	P
3.5.4	The documentation shall have no usage of legalese or other complicated technical terms or jargon that would confuse a user.	Documentation	Documentation	T.3.5.4	P
3.5.5	The documentation should explain all parts of the software the user can interact with.	Documentation	Documentation	T.3.5.5	F
3.6.1	The degree of GPS precision shall be 6 decimal places for latitude and longitude. For example, (28.6024 N, 81.2001 W) which are the GPS coordinates for the University of Central Florida. The accuracy of the GPS data will vary depending location, weather and the number of satellites in line of sight to the GPS hardware on the mobile device. The Google Maps API will be used to facilitate this requirement.	Database, Retrieves Database Information	Documentation	T.3.6.1	F
3.6.2	GPS coordinates in the form of latitude and longitude for each individual employee must be retained in the database.	Database, Retrieves Database Information	Firebase Database, Algorithm	T.3.6.2	F

3.6.3	The employee's name and unique ID will need to be retained together in the database.	Database, Make changes to the database.	Database	T.3.6.3	F
3.6.4	The date and time of the gathered GPS coordinates shall be retained alongside the rest of the gathered data in the database.	Database	Database, User,	T.3.6.4	F
3.6.5	Logins shall be retained in the database.	Client-Server, Repository	Database, Login	T.3.6.5	F
3.7.1	A database shall be required for the backend to provide long term storage for our client's data. As our client's grow in number the costs associated with the Firebase database platform we chose will grow as well.	Database	Firebase Database	T.3.7.1	P
3.7.2	A database administrator(s) will be needed to perform maintenance on the system as required. The system may need to be debugged at times and this action will require skilled personnel.	Database	Firebase Database	T.3.7.2	P
3.7.3	The Employee Tracker application shall require the client to have access to computers and the client's employees to have access to mobile devices. We will not need to provide any hardware to the clients for the application to function as intended. In addition, we will not require physical space or supporting amenities as the client's business will cover those aspects of their operation.	Subscriber	Web Application, Database	T.3.7.3	P
3.8.1	Access to the system shall be controlled and secured by logins for each client.	Server-Client	Firebase Database	T.3.8.1	P
3.8.2	Client's data shall be isolated from other clients through logins.	Server-Client	Firebase Database	T.3.8.2	P
3.8.3	Client's data shall be encrypted with 256-bit AES standard	Database	Firebase Database	T.3.8.3	F
3.8.4	Database shall be tested against SQL injection	Database	Firebase Database	T.3.8.4	P

3.8.5	Automatic backups shall be performed once per day offsite (Firebase cost dependent). Backups are off site on Google's servers and protected from water, fire and other natural disasters. Data can be recovered if needed at any time.	Database	Firebase Database	T.3.8.5	F
3.8.6	Employee's shall not spoof their location.	Server-Client	Mobile Application	T.3.8.6	F
3.9.1	The Employee Tracker Application shall be available as a web application and mobile application.	Client-Server, Repository	Web Application, Mobile Application	T.3.9.1	F
3.9.2	Software shall be maintained by us, the software creators.	Repository	Employee Tracker	T.3.9.2	P
3.9.3	Automation of bug-detection will be in place	Client-Server, Repository, Database	Firebase Database, UI, Web/Mobile Applications	T.3.9.3	F
3.9.4	Downtime must be minimal (less than 30 minutes a week during working hours), Response time to issues should be within the hour of the issue arising.	Database	Connection, Firebase Database	T.3.9.4	F

Conclusion

The system has not been finished so it is hard to give a final conclusion on the product, but what can be said is that the product needs some work still. Testing will definitely be required throughout more of our development and reviewing our table of tests-to-be-done, it seems as though more testing is unnecessary. We are on a projected path to be done on time, so the product cannot be currently delivered. Maintenance will not be an issue as the system is fairly simple so any complications can be quickly addressed.

As to what remains to be done, we are currently working on the Google Maps API as well as the functionality for GPS tracking since they are one in the same. We also need to implement our automated testing to ensure structure and reliability of our system to the customer. Our development environment and our Firebase server are well under way and supply a solid foundation for us to efficiently finish the rest of what needs to be done.