Software Requirements Specification

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

Section 1: Introduction

Software to be Produced:

- The software being produced will consist of a web application and a mobile application, both part of a system to enable GPS tracking of the mobile application users. Management and the tracking portion of the system will be accessed through the web interface by management, while the mobile application will be used by employees to sign in and enable tracking. Please refer to the Concept of Operations document, specifically the Proposed System section.

Applicable Standards

- The system requirements standards can be found under the Applicable Standards section of the Project Management Plan document.

Definitions, Acronyms, and Abbreviations

- ReactJS/React Native: Javascript framework which will be the core of our applications.

- Flask: A Python web microframework which enables the creation of RESTful APIs which will connect the applications to the database.

- Firebase: An online mobile development suite that will serve as the backend database and management for the system.

Section 2: Product Overview

Assumptions:

- Assume that the web-application users will be using the latest versions of Google Chrome, Mozilla Firefox, or Internet Explorer as their web browser.

- Assume that the mobile application users will be using the latest versions of Android or iOS on their mobile device.

- Assume that the hardware specifications of the device being used match the requirements of the operating system, ensuring capability to run the mobile application efficiently.

- Assume that React and Flask frameworks will have the ability to meet required functions of the software.

Stakeholders:

- Clients - Includes Professor Nassif, Salih Safa Bacanli , and Jun Xu. This group is essentially paying for the software to be developed. This group is interested in the outcome of the project as they are funding the it and will sell it once they own the software.

- Customers - This is a potential company or group that would purchase the software from our clients after the software is developed. This group is interested in the outcome of the project as they will be paying for it and implementing it into their practices.

- Users - This would be the employees of this potential company that has bought the software, including a group of managers and a group of employees that will be tracked. This group is interested in the outcome of the project as they will be using this software personally. They will be affected directly by the quality of the software.

- Software Engineers - This would be our team who is creating documentation and developing the software that will be sold to the client. Specifically, this will be the members listed at the top of the document. This group is interested in the outcome of the project as they are developing the project to meet the client's requirements. If they do not meet these requirements, they may face consequences such as termination of employment.

Event Table:

Event Name	External Stimuli	External Responses	Internal data and state
Employee Unsuccessful Login	An employee attempts to log into the system through the mobile application by taking a picture of themselves and including their username and password.	Employee is shown an message, "Invalid Login", regarding the failure of their verification on the system.	EmployeeLoginAttempt -> sends credentials to server and database. EmployeeVerification -> returns false/failure to login or verify user. isLoggedIn -> stays set to false
Employee Sucessful Login	An employee attempts to log into the system through the mobile application by taking a picture of themselves and including their username and password.	Employee is shown an message, "Logged In", regarding the success of their verification on the system, and tracking is enabled. If it is their first time logging into the system by using the one time password generated for the user, they are prompted to submit a new password.	EmployeeLoginAttempt -> sends credentials to server and database. EmployeeVerification -> returns true/success to login or verify user. Verification keys are also cached locally for the user passwordReset -> if the user logs into the system for the first time, this was set to true and the new password provided by the user is sent to the server, hashed, salted, and stored for future verification. This is then set to false for the user. isLoggedIn -> is set to true in the server and application

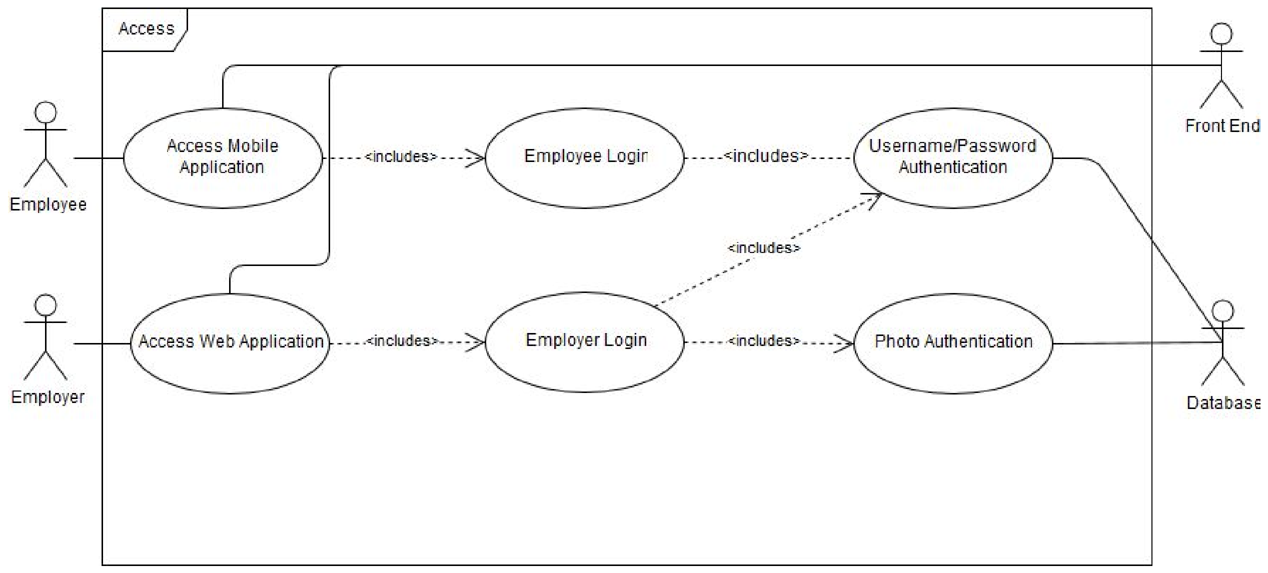
Employer/Management Unsuccessful Login	The employer attempts to log into the system through the web application by their username and password.	Employer is shown an message, “Invalid Login”, regarding the failure of their verification on the system.	EmployerLoginAttempt -> sends credentials to server and database. EmployerVerification -> returns false/failure to login or verify user. isLoggedIn -> stays set to false
Employer/Management Successfull Login	The employer attempts to log into the system through the web application by their username and password.	Employer is shown an message, “Logged In”, regarding the success of their verification on the system.	EmployerLoginAttempt -> sends credentials to server and database. EmployerVerification -> returns true/success to login or verify user. Verification keys are also cached locally for the user isLoggedIn -> is set to true in the server and application
Employee Physically Moves/GPS Location Periodically Updates	The mobile application sends the location of the device/user to the server every 10 minutes.	Employer web application gets updated with the new location of the employee	EmployeeLocationVerification-> the employee location is verified on the mobile application through checks to see if Location Faking is enabled in the device settings, or that the device is rooted or jailbroken. If the device meets any of those requirements, then an alert gets sent to the server and the management stating that the device is untrusted. EmployeeLocation -> location sent by the mobile application gets stored in the server database, along with all past locations of that user. EmployeeLocationView -> The employer/management application receives the GPS and Date updates from the server/database and displays them. EmployeeVerification -> verification keys cached locally for seamless reconnection are revalidated against the server.
Loss of Connection from Mobile Application	The mobile application loses service/internet connectivity/ability to communicate with server.	The mobile application shows a notification message that “Connectivity to the server has been lost. Attempting reconnection”. If 30 minutes of	EmployeeVerification -> verification keys cached locally for seamless reconnection is revalidated against the server. EmployeeLocation -> locations that would be sent to the server

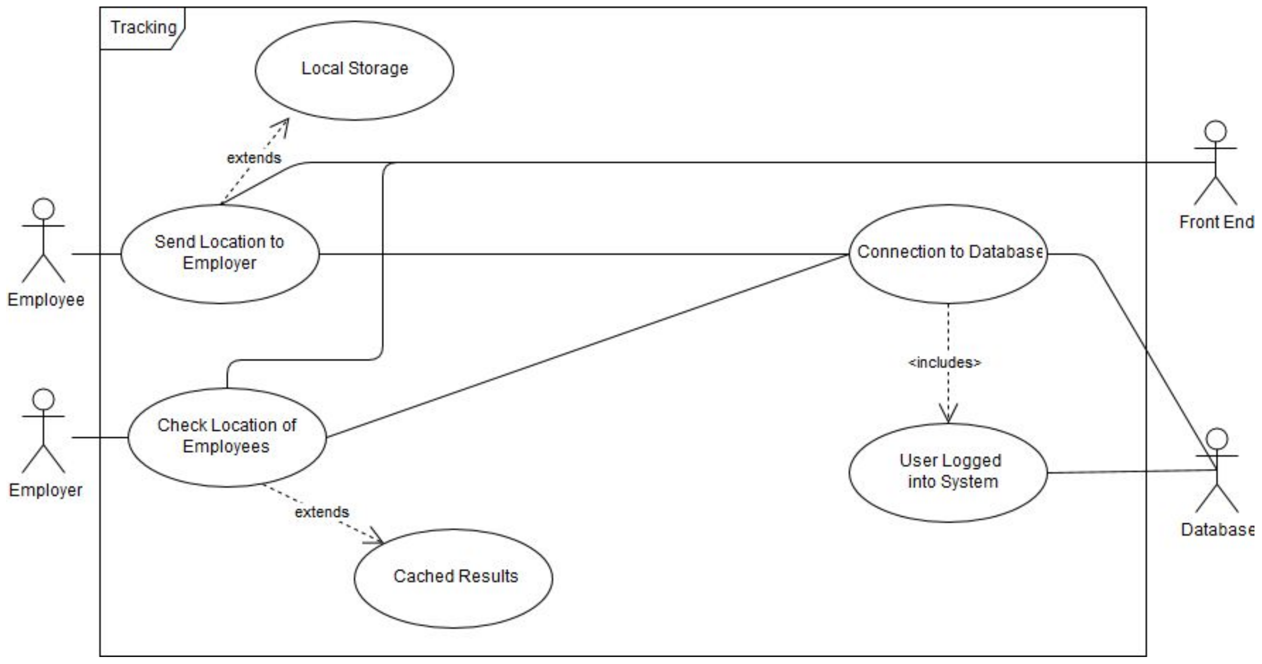
		no connection continues, then the user is logged out.	get stored locally on the mobile device until communication with the server is restored, at that point the list of locations is sent by the mobile application. EmployeeLocationView -> The management dashboard shows that the mobile application for the respective employee has not been updated in the expected timeframe of 10 minutes. isLoggedIn -> stays set to true in the server and application, until a timeframe of 30 minutes of no connectivity passes, then they are logged out and the status is set to false, and local verification cache is invalidated.
Loss of Connection from Web Application	The web application looses service/internet connectivity/ability to communicate with server.	The web application shows a notification message that "Connectivity to the server has been lost. Attempting reconnection". If 30 minutes of no connection continues, then the user is logged out.	EmployerVerification -> verification keys cached locally for seamless reconnection is revalidated against the server. EmployeeLocationView -> the last received locations of tall employees are cached for viewing on the system. isLoggedIn -> stays set to true in the server and application, until a timeframe of 30 minutes of no connectivity passes, then they are logged out and the status is set to false, and local verification cache is invalidated.
Employer Adds User	The employer adds an employee "[new user]" to the system through the web application	The web application shows a message asking if the information inputted is correct, and if it is, then the web application shows "Success: User [new user] is added to the system"	newEmployee -> a new employee is added to the database of employees, and a one time password is generated for the new user to log into the system and reset their password. EmployerVerification -> verification keys are revalidated to increase security of the system. isLoggedIn -> set to false
Employer Removes Existing User	The employer removes an employee "[old user]" from the system through the web application	The web application shows a message asking if the information inputted is correct and that the removal of the user will be final. If it is, then the web	removeEmployee -> the employee is removed from the system, along with all data pertaining to the employee

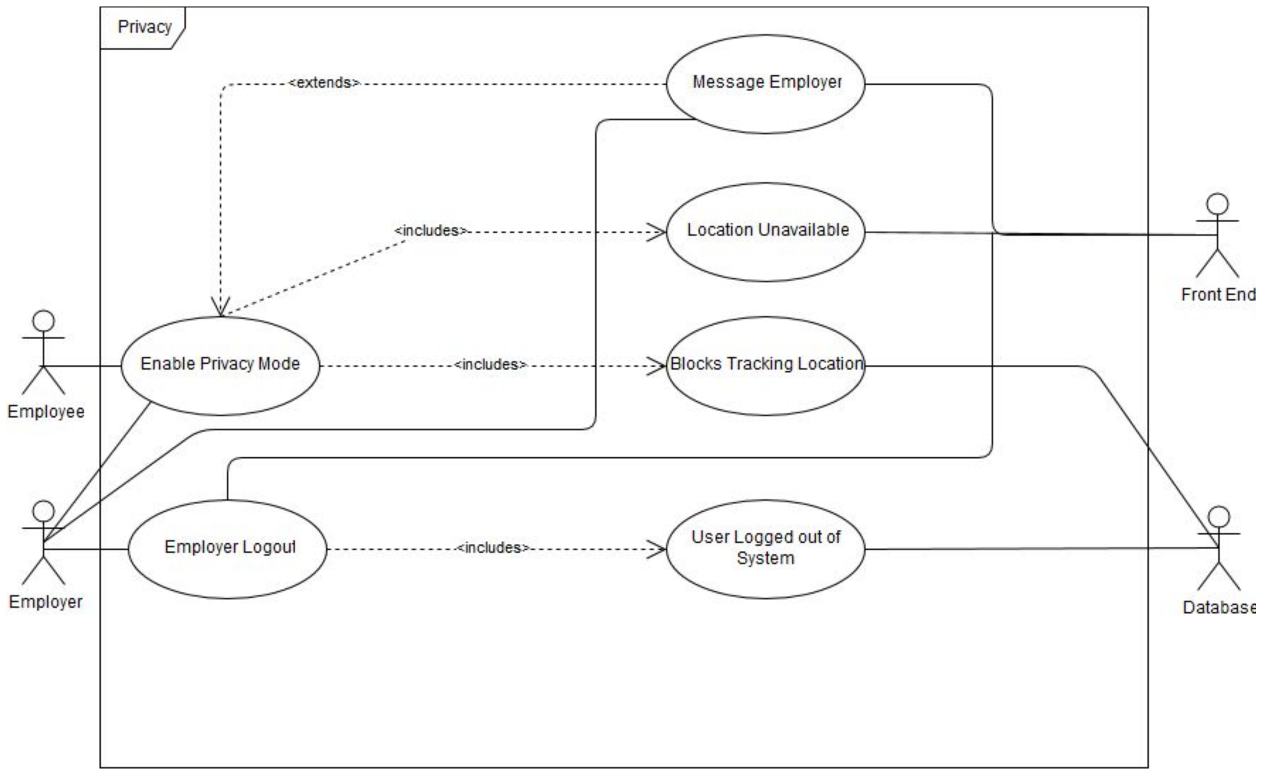
		application shows “Success: User [old user] is removed to the system”. If [old user] is still logged in, the system will logout the user from the web application.	EmployeeVerification -> is set to false/unverified and the existing employee is logged out of the system if they are currently logged in. EmployerVerification -> verification keys are revalidated to increase security of the system. isLoggedIn -> set to false for the old user.
Employer Removes Non Existing User	The employer removes an employee “[old user]” from the system through the web application	The web application shows a message asking if the information inputted is correct and that the removal of the user will be final. The web application shows “Error: User [old user] does not exist in the system”.	EmployerVerification -> verification keys are revalidated to increase security of the system. removeEmployee -> returns with an error stating that the employee is not found in the database.
Employee Enables Privacy Mode	The employee enables privacy mode on the app, which disables location tracking. They also have the option to send a message stating why they enabled the mode.	The message is sent to the employer/management accounts on the web application, and the user's last known location is still displayed with a icon denoting that they are in privacy mode.	privacyMode -> is set to true on the device, and the status and optional message is sent to the employer web application. EmployeeLocation -> is disabled when privacyMode is enabled. isLoggedIn -> stays set to true.
Employee/Employer Logs out of the System	The employer or employee logs out of the system by clicking on a logout button	The web or mobile application displays a message that the user is successfully logged out of the system.	isLoggedIn -> gets set to false. EmployeeVerification -> local cache is invalidated immediately. EmployerVerification -> local cache is invalidated immediately.
Employer Lookup of Valid Employee	The employer searches for an employee ID number and date through the web application	The web application will display the user and a list of their locations on the date given.	EmployeeLocationView ->is set to show the locations of the user searched for.
Employer Lookup of Invalid Employee	The employer searches for an employee id number and date through the web application	The web application will display an error stating that the employee id or the date are invalid.	EmployeeLocationView ->is set to show nothing, and returns error stating that the date or employee ID number cannot be found
Employee Starts Mobile Application for the First Time	The employee opens the mobile application for the first time.	The mobile application will request data, location, and camera access from the user. If the user chooses to not allow	acceptedPermission -> is set to false if the employee does not accept the permission requests

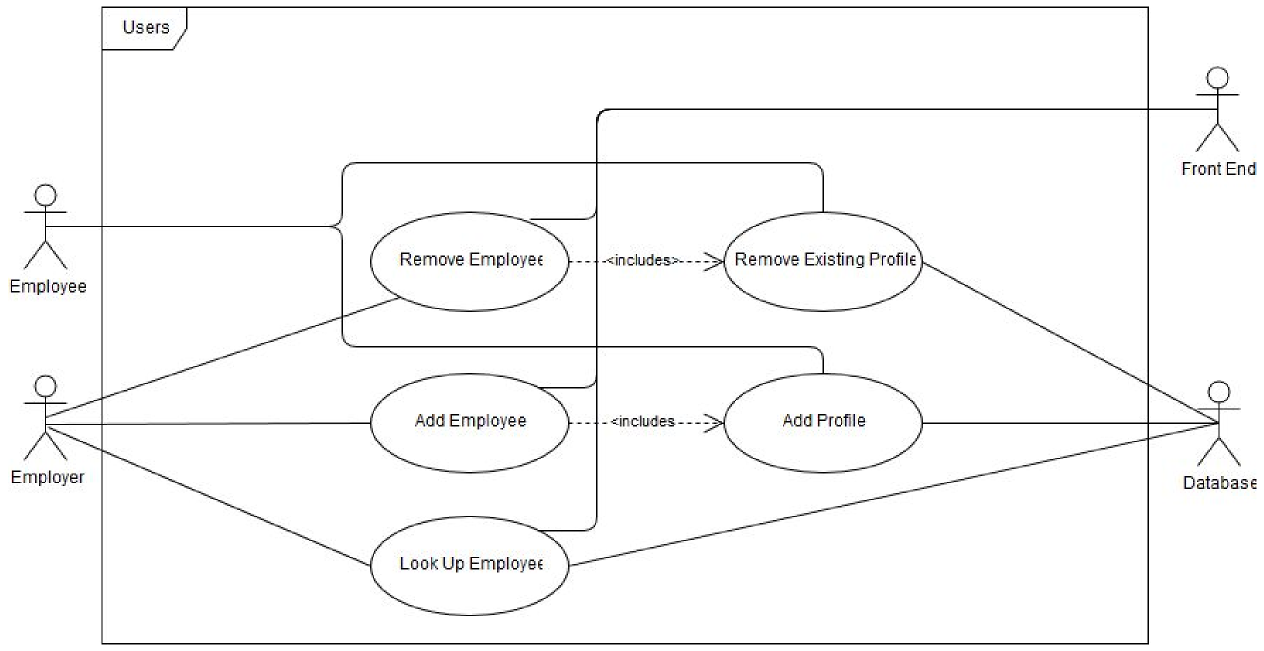
		any of those requirements, then the app will show a message explaining that those accesses are necessary for the application to function, and to contact your employer with any further questions.	of the app, otherwise is set to true. This gets sent to the server.
Employee Starts Mobile Application, and Required Services do not Function.	The employee opens the mobile application and attempts to log in, however the GPS, camera, or data connection are nonfunctional.	The application will display a user friendly error stating that one of the required functions of the phone are unavailable and will state that they cannot log into the system, and to contact your employer with any further questions.	acceptedPermission -> is set to false in case of a error on the device. This gets sent to the server.

Use Case Diagrams:




Tracking
Local Storage
extends
Send Location to Employer
Employee
Employer
Check Location of Employees
extends
Cached Results
Front End
Connection to Database
<includes>
User Logged into System
Database


Privacy
<extends>
Message Employer
<includes>
Location Unavailable
Front End
Employee
Enable Privacy Mode
<includes>
Blocks Tracking Location
Employer
Employer Logout
<includes>
User Logged out of System
Database



Use Case Descriptions:

- Access
    - Employees can access the front end of the mobile application by logging into the system, which requires the usage of the database. The process of logging into the system includes username/password authentication as well as photo authentication.
    - Employers can access the front end of the web application by logging into the system, which requires the usage of the database. The process of logging into the system includes username/password authentication.
- Tracking
    - Employees will send their location to the employer using the front end of the application and the database. If the user is logged in and connected to the database, the user's location will be stored and updated in the database. If there is no connection to the database, local storage can be used to store the location and send later when connection is reestablished.
    - Employers can check the location of employees using the front end of the application and the database. If the user is logged in and connected to the database, the employer can check the locations of each employee stored in the database. If there is no connection to the database, the employer can check previous locations that have been stored in the cache. The locations will update to be current when the connection is reestablished.

- Privacy
    - Employees can enable privacy mode, which will block tracking from the database and make the location unavailable on the front end. When privacy mode is enabled, employees will also have the ability to send a message to their employer, describing the reasoning for enabling privacy mode.
    - Employers can logout if desired, which will remove access to the web application until the employer logs in again. They will also be unavailable from the front end of the application.
- Users
    - Employers can lookup, add, and remove users in the database. Adding and removing users will reflect on the front end and will also either enable or disable the user to use the system. Lookup will aid the Employer in finding information about existing profiles in the database.

Section 3: Specific Requirements

3.1 Functional Requirements

Table No: 3.1 **Functional Requirements**
No: 3.1.1
Statement: The Employee Tracker shall allow the Employee to log in, update location, and see other users using the mobile application.
Source: The engineering team
Dependency: 3.1.2 Functional requirement, 3.2.1 Interface requirement
Conflicts: None
Supporting Materials: None
Evaluation Method: If the application allows you to log in with out “invalid login” and lets you update your current location.
Revision History: Navon Francis, 10-15-2017, Initial
No: 3.1.2

Statement: The Employee Tracker shall allow the Manager to see updates from the employee on the web application.
Source: The engineering team
Dependency: 3.1.1 Function requirement, 3.2.1 Interface requirement
Conflicts: None
Supporting Materials: None
Evaluation Method: If the application allows you to log in without “invalid login” and allows you to access GPS locations of employees.
Revision History: Navon Francis, 10-15-2017, Initial
No: 3.1.3
Statement: The login username and password will be verified to be valid according to the requirements set by the company.
Source: The Engineering Team
Dependency: 3.2.1
Conflicts: None
Supporting Materials: Security rules set by the company utilizing the system
Evaluation Method: Through testing the system, usernames and passwords that contain valid characters as dictated by the company using the system should pass, and all other tests that contain invalid characters should fail.
Revision History: Ryan Hoeck, 10-15-2017, Initial.
No: 3.1.4

Statement: 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.
Source: “After login, GPS location of the employee will be tracked automatically by the system and send to the admin every 10 minutes.”
Dependency: 3.1.1, 3.1.3, 3.2.2, 3.1.4
Conflicts: None
Supporting Materials: Concept of Operations, Project Management Plan
Evaluation Method: The system can be tested to verify that when loss of connection occurs, the mobile application continues to collect location data for up 30 minutes if internet connectivity has not resumed by that point. Upon 30 minutes of no connection and the user is still logged into the system, a notification will be sent to the employer stating that the employee's device has not had a connection for 30+ minutes.
Revision History: Ryan Hoeck, 15-10-2017, Initial.
No: 3.1.5
Statement: The mobile application will notify the user and the employer if the mobile device’s GPS or camera malfunctions.
Source: Concept of Operations, Project Management Plan, Software Design Team
Dependency: The mobile device running the mobile app has a data connection.
Conflicts: None
Supporting Materials: Concept of Operations, Project Management Plan
Evaluation Method: The system can be tested to ensure the notification of failure to interact with either the GPS or camera of the device is sent to the employer and the user.
Revision History: Ryan Hoeck, 15-10-2017, Initial.

No: 3.1.6
Statement: The mobile application will send the locally stored locations to the server upon successful reconnection.
Source: “After login, GPS location of the employee will be tracked automatically by the system and send to the admin every 10 minutes.”
Dependency: 3.1.4
Conflicts: None
Supporting Materials: Concept of Operations
Evaluation Method: The mobile application can be tested to ensure that on disconnection and reconnection to the internet that locations were stored in the timeframe between the two events, and that the list of locations gets successfully sent to the server.
Revision History: Ryan Hoeck, 15-10-2017, Initial.

3.2 Interface Requirements

Table No: 3.2 **Interface Requirements**
No: 3.2.1
Statement: The Employee Tracker application shall check username and password for each user log in. 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.
Source: The engineering team
Dependency: 3.1.1, 3.1.2, 3.1.3
Conflicts: User can get an “Invalid Login” when the request has been made to validate with Firebase. Then the username or password was entered incorrectly.
Supporting Materials: None

Evaluation Method: The username and password should be verified as strings with characters that are allowed regarding dependency 3.1.3. These should then be verified as successfully being sent and validated with Firebase.
Revision History: Navon Francis, 10-15-2017, Initial. Ryan Hoeck, 10-15-2017, Added specific formats of data.
No: 3.2.2
Statement: The Employee Tracker application shall fetch, get (fetch method), and post requests to and from Firebase to validate against the frontend that what the user input has integrity.
Source: The engineering team
Dependency: 3.2.1 Functional requirement
Conflicts: Assuming the input data (ex: username and password) is correct, reactJS or Firebase may come out with a critical update crippling the way we currently send information.
Supporting Materials: None
Evaluation Method: ReactJS and React Native use an asynchronous Fetch() method that will be used to send inputs such as, username [Text-String], password [Text-String], GPS coordinates [Geographical point (latitude, longitude)], Timestamp [Date and time], checked-in [Boolean], privacy settings [Boolean]. Each input must be critically accurate to not misrepresent the employees or managers.
Revision History: Navon Francis, 10-15-2017, Initial
No: 3.2.3
Statement: 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.
Source: “After login, GPS location of the employee will be tracked automatically by the system and send to the admin every 10 minutes”
Dependency: 3.1.3, 3.1.1
Conflicts: None

Supporting Materials: Concept of Operations
Evaluation Method: The system can be tested to verify that when a user logs in, their location is sent to the database, and their location is continuously sent to the database every 10 minutes. If their location is not sent timely, then this fails. The location should have 6 points of precision, such as 28.6024° N, 81.2001° W and no more or less, such as 28.60° N, 81.20° W.
Revision History: Ryan Hoeck, 15-10-2017, Initial.
No: 3.2.4
Statement: All data transfers between the Firebase server and the web or mobile application shall be through a RESTful API.
Source: Software Development Team
Dependency: 3.1.3, 3.1.1
Conflicts: None
Supporting Materials: Concept of Operations
Evaluation Method: The system can be tested to verify that all user updates to and from the Firebase server are sent through a REST API, the fields of it being gpsLocation, username, password, coordinateList, loggedInStatus, timestamp in ISO standard, privacySettingStatus.
Revision History: Ryan Hoeck, 15-10-2017, Initial.

3.3 Physical Environment Requirement

Table No: 3.3 Physical Environment Requirement
Statement: No physical environment is required.

3.4 User and Human Factors Requirement

Table No: 3.4 **User and Human Factors Requirement**
No: 3.4.1 User and Human Factors Requirements

Statement: The Employee Tracker Application shall have these 3 user groups: The Employer, Employee, and Admin.
Source: Software Development Team
Dependency: None
Conflicts: None
Supporting Materials: Concept of Operations
Evaluation Method: There will be 3 distinct user profiles with individual capabilities.
Revision History: Marc Simon, 10-15-2017, Initial
No: 3.4.2
Statement: The user groups shall have these capabilities: **Employer** - Capable of tracking employees hours, lunch breaks, and receive notifications from employees. **Employee** - Capable of controlling privacy policies, turn tracking off, and clock in and out for work. **Admin** - Full control over the system with the ability to act as an employer or employee for dispute resolution, general customer service and maintenance of the system.
Source: Software Development Team
Dependency: 3.4.1
Conflicts: None
Supporting Materials: Use Case Diagram
Evaluation Method: The software will be tested in each profile for their respective capabilities. If any capability does not work then this requirement is unsatisfied.
Revision History: Marc Simon, 10-15-2017, Initial
No: 3.4.3

Statement: The Employee Tracking App shall prevent misuse by the user groups
Source: Software Development Team
Dependency: 3.4.1, 3.4.2
Conflicts: None
Supporting Materials: None
Evaluation Method: Alerts will be generated when employees attempt to spoof their locations. Employees will be able to control their privacy policies and turn off GPS tracking. Admin will have full control of system to mitigate and resolve disputes and issues regarding misuse and maintenance.
Revision History: Marc Simon, 10-15-2017, Initial
No: 3.4.4
Statement: 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.
Source: Software Development Team
Dependency: None
Conflicts: None
Supporting Materials: None
Evaluation Method: Employees and Employers receive the proper documents and training for the systems. The fonts throughout the user interface must be at least 11pt in size.
Revision History: Marc Simon, 10-15-2017, Initial. Ryan Hoeck, 10-15-2017, Added specifics to the user requirements and what abilities they should possess.

3.5 Documentation Requirements

Table No: 3.5 **Documentation Requirements**

No: 3.5.1
Statement: The documentation shall be stored with the software and updated in accordance to functional changes of the software.
Source: The Development Team
Dependency: None
Conflicts: None
Supporting Materials: None
Evaluation Method: Every time a functional requirement is updated or changed, any relevant user documentation needs to be updated as well.
Revision History: Ryan Hoeck, 15-10-2017, Initial.
No: 3.5.2
Statement: The documentation will have pictures and diagrams of the software, with footnotes explaining what each button does.
Source: The Development Team
Dependency: None
Conflicts: None
Supporting Materials: None
Evaluation Method: The final documentation should have diagrams and images of the working software, with explanations of what each user interface element does.
Revision History: Ryan Hoeck, 15-10-2017, Initial.
No: 3.5.3
Statement: The documentation should be concise and not span hundreds of pages.

Source: The Development Team
Dependency: None
Conflicts: None
Supporting Materials: none
Evaluation Method: The final documentation given to the user should not exceed 100 pages. Subsections throughout the document should be concise, explaining the functionality of the section without excess text.
Revision History: Ryan Hoeck, 15-10-2017, Initial.
No: 3.5.4
Statement: The documentation shall have no usage of legalese or other complicated technical terms or jargon that would confuse a user.
Source: The Development Team
Dependency: None
Conflicts: None
Supporting Materials: None
Evaluation Method: The final documentation should have no complicated technical terms, this can be evaluated by rereading and providing explanations to any necessary technical jargon.
Revision History: Ryan Hoeck, 15-10-2017, Initial.
No: 3.5.5
Statement: The documentation should explain all parts of the software the user can interact with.
Source: The Development Team
Dependency: None

Conflicts: None
Supporting Materials: None
Evaluation Method: There should be a documentation section for each user facing function in the software.
Revision History: Ryan Hoeck, 15-10-2017, Initial.

3.6 Data Requirements

Table No: 3.6 **Data Requirements**
No: 3.6.1
Statement: 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.
Source: The employee’s location will be tracked.
Dependency: The employee must exist in the database. The employee grants permission to access their location on their mobile device. Internet connectivity is available on their mobile device.
Conflicts: None
Supporting Materials: None
Evaluation Method: The employee’s location will be visible to the employer.
Revision History: Brandon Bradley, 10/15/17, Initial
No: 3.6.2
Statement: GPS coordinates in the form of latitude and longitude for each individual employee must be retained in the database.
Source: Each employee must be tracked individually.

Dependency: The employee must exist in the database.
Conflicts: None
Supporting Materials: None
Evaluation Method: The employee’s last known location will be visible to the employer.
Revision History: Brandon Bradley, 10/15/17, Initial
No: 3.6.3
Statement: The employee’s name and unique ID will need to be retained together in the database.
Source: Each employee must be tracked individually.
Dependency: The employee must be initially added to the database.
Conflicts: None
Supporting Materials: None
Evaluation Method: An employee lookup retrieves a valid employee from the database.
Revision History: Brandon Bradley, 10/15/17, Initial
No: 3.6.4
Statement: The date and time of the gathered GPS coordinates shall be retained alongside the rest of the gathered data in the database.
Source: Employer must know where and when which defines an employee’s location.
Dependency: The employee must exist in the database. The employee grants permission to access their location on their mobile device. Internet connectivity is available.
Conflicts: None
Supporting Materials: None

Evaluation Method: A timestamp is presented when viewing an employee’s last known location.
Revision History: Brandon Bradley, 10/15/17, Initial
No: 3.6.5
Statement: Logins shall be retained in the database.
Source: Employer must login to view employee’s location. Super admin must login to perform database maintenance.
Dependency: The employer must be initially added to the database. The database administrator must be initially added to the database. Internet connectivity is available.
Conflicts: None
Supporting Materials: None
Evaluation Method: Employer can login to view employees’ locations. Super admin can login to perform database maintenance.
Revision History: Brandon Bradley, 10/15/17, Initial

3.7 Resource Requirements

Table No: 3.7 **Resource Requirements**
No: 3.7.1
Statement: 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.
Source: Client’s data must be stored to be retrieved by the application.
Dependency: None

Conflicts: None
Supporting Materials: None
Evaluation Method: This is a functional requirement. All other requirements depend on this requirement being fulfilled.
Revision History: Brandon Bradley, 10/15/17, Initial
No: 3.7.2
Statement: 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.
Source: The database will need to be constantly available and thus free of bugs and consistently maintained.
Dependency: The database as a functional requirement exists.
Conflicts: None
Supporting Materials: None
Evaluation Method: The database administrator is able to maintain the database
Revision History: Brandon Bradley, 10/15/17, Initial
No: 3.7.3
Statement: 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.
Source: The web and mobile application must be accessed through a mobile or desktop computer.
Dependency: None
Conflicts: None

Supporting Materials: None
Evaluation Method: The client can access the developed mobile and web applications.
Revision History: Brandon Bradley, 10/15/17, Initial

3.8 Security Requirements

Table No: 3.8 **Security Requirements**
No: 3.8.1
Statement: Access to the system shall be controlled and secured by logins for each client.
Source: Employers must be able to view employee's location.
Dependency: Database is initially created.
Conflicts: None
Supporting Materials: None
Evaluation Method: The employer can login and view **only** their employee's locations.
Revision History: Brandon Bradley, 10/15/17, Initial
No: 3.8.2
Statement: Client's data shall be isolated from other clients through logins.
Source: The employer can login and view **only** their employee's locations.
Dependency: Database is initially created. Client logins have been created.
Conflicts: None
Supporting Materials: None
Evaluation Method: The employer can login and view **only** their employee's locations.

Revision History: Brandon Bradley, 10/15/17, Initial
No: 3.8.3
Statement: Client's data shall be encrypted with 256-bit AES standard
Source: Concept of Operations
Dependency: Database is initially created. Client logins have been created.
Conflicts: None
Supporting Materials: None
Evaluation Method: Confirmed retrieval of data with encryption key.
Revision History: Brandon Bradley, 10/15/17, Initial
No: 3.8.4
Statement: Database shall be tested against SQL injection
Source: Concept of Operations
Dependency: Database is initially created. Test logins have been created.
Conflicts: None
Supporting Materials: None
Evaluation Method: SQL injection will be tested against Firebase's built in security and validation checks.
Revision History: Brandon Bradley, 10/15/17, Initial
No: 3.8.5

Statement: Automatics 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.
Dependency: Firebase's automatic backups are free. This project is not funded.
Conflicts: None
Supporting Materials: None
Evaluation Method: Observe Gzip compressed backup files through Firebase
Revision History: Brandon Bradley, 10/15/17, Initial
No: 3.8.6
Statement: Employee's shall not spoof their location.
Dependency: Database is initially created. Employee exists in database. Employee has granted permission for location tracking.
Conflicts: None
Supporting Materials: None
Evaluation Method: Employee's mobile device will be checked for root access on Android and jailbreak installation on iOS. This will serve as a red flag but not necessarily a confirmation of spoofing. Location will be checked against previous location for travel time or distance discrepancies.
Revision History: Brandon Bradley, 10/15/17, Initial

3.9 Quality Assurance Requirements (AKA nonfunctional, Quality Requirements)

Table No: 3.9 **Quality Assurance Requirements Requirements** (Nonfunctional Requirements)
No: 3.9.1
Statement: The Employee Tracker Application shall be available as a web application and mobile application.

Source: Concept of Operations
Dependency: 3.1, 3.2, 3.6
Conflicts: If functionality, interface, or data management stop being available or working, there will be conflicts with this requirement
Supporting Materials: None
Evaluation Method: Employees will be able to clock in and out using the mobile app and Employers will be able to check location data, employee hours, and other relevant data using mobile and web application.
Revision History: Marc Simon, 10-15-2017, Initial
No: 3.9.2
Statement: Software shall be maintained by us, the software creators.
Source: Project Management Plan
Dependency: None
Conflicts: None
Supporting Materials: None
Evaluation Method: Development team will sustain system, fix bugs and errors shown on error reports, and deal with any privacy concerns.
Revision History: Marc Simon, 10-15-2017, Initial
No: 3.9.3
Statement: Automation of bug-detection will be in place
Source: Project Management Plan
Dependency: bug-detection code works

Conflicts: If there is a bug in the bug detection, then it will not be automated, but will be handled by the software team
Supporting Materials: none
Evaluation Method: After each implementation of revisions and updates occurs, our program will tell us whether or not the system is in good condition or not. Otherwise, the product will still be working and can be tested by use of the customer.
Revision History: David Simoneau <10-15-17> initial
No: 3.9.4
Statement: 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.
Source: Project Management Plan
Dependency: Code is capable of running for extended periods of time without issues., 3.9.2
Conflicts: Downtime and response time could be extended if any of the software team is unavailable or otherwise unable to be contacted.
Supporting Materials: none
Evaluation Method: Any issue that arises in a production environment should be timed and responded to within an hour, and any downtime during work/performance hours should not exceed 30 minutes.
Revision History: Ryan Hoeck, 10-15-17, initial

Section 4: Supporting Material

- Reference the Concept of Operations and Project Management Plan for explanations of the above requirements.
- The Software Development Team decided that a React Native/React JS app would be most beneficial for the development platform of the project, and communication between the applications should be through a REST API. Also, throughout discussions the various other requirements were decided.
- The Software Design Description given also lead to these requirements stated.

- Lectures from the Class (COP 4331) and the associated Lab section were taken into consideration when creating these requirements.