Capstone Intelligence
System Design

Group 3

September 22, 2022

Samuel Crawford
Joshua Guinness
Hassan Jasim
Nicholas Mari

Contents

1 Revisions

#	Author(s)	Description	Date
Rev 0	All	Created first draft of document	December 10, 2021
Rev 0.1	Samuel	Began work on MIS and adding diagrams	December 29, 2021
Rev 0.2	Joshua	Scope, Purpose, MIS for Milestone tracking	January 2, 2022
Rev 0.3	All	Discussion of format and design; preliminary work	January 2, 2022
Rev 0.4	All	Continuing work from Rev 0.3	January 3, 2022
Rev 0.5	All	Continuing work from Rev 0.3 and finishing touches	January 4, 2022
Rev 0.5.1	Samuel	Updated specification based on design changes during implementation	January 23, 2022
Rev 0.5.2	Samuel	Added assumptions from other documents	February 23, 2022
Rev 1.0	Hassan	Updated Component Diagram	February 24, 2022
	Samuel	Updated Module Traceability; added Site Information module; made minor changes to match current implementation	February 24, 2022
	Nicholas, Samuel	Updated Exception Handling sections; minor changes	February 28, 2022
	Joshua	Updated Database and Authentication Modules; minor changes	February 28, 2022
	Joshua	Updated Equipment data type, and Site Information Module based on database schema changes	March 29, 2022

2 Purpose

2.1 System Purpose

The purpose of the system is a software solution that will help the supervisors and assistant supervisors of the general contractors manage their jobsite(s) and better understand what is happening onsite for both day to day and long-term time frames. This software solution will support daily reporting, scheduling and organization of onsite deliveries, tracking of milestones, construction equipment tracking, and potentially data analysis.

The system will be used by the supers and assistant supers on the jobsite. It will be used as they walk around the jobsite, as well as from their makeshift office on site. It will also be used at the head office of the general contractor.

2.2 Document Purpose

The purpose of this document is to illustrate and justify the decomposition of the system into its components and provide a Modular Interface Specification for each component in the system. This document will serve as the basis for the implementation work to be completed in the coming months.

3 Scope

This system is designed for use on construction sites by their supervisors. It includes the application itself and the database used to store its data.

3.1 Assumptions

This application is designed with the following assumptions:

A1. Site supers and assistant supers will have access to the internet through WiFi set up in their site trailers or through their companies data plan for the majority of their time on site.

A2. This app will primarily be used by general contractors and the subcontractors they hire.

A3. The average super and sub is not as tech-savvy as the average citizen.

A4. The average age of users of this app will be around 45-55 years old.

A5. Work will not be done on multiple milestones in parallel.

Rationale: Milestone tracking is a feature with no user feedback, and since it is very uncommon for more than one milestone to be worked on concurrently, simplicity will be prioritized.

A6. The language being used for implementation has standard Date, Time, and DateTime types.

A7. Each delivery will be associated with a maximum of one subcontractor and one piece of equipment.

A8. No delivery will take place across multiple days.

A9. Each user can be uniquely identified by their email address.

4 Project Overview

4.1 Normal Operation

This application is to be used by site supers in order to properly track their job site. Supers will use the application on their phones, which will allow them to use the product while walking around the job site. This will ensure that all information about their job site is available in a single package, that they can reference through the application without needing to rely on memory. The data about subcontractors and the history of the job site tracked through the application will allow supers to make better decisions at their job sites to ensure timely completion of their construction.

4.2 Behaviour Overview

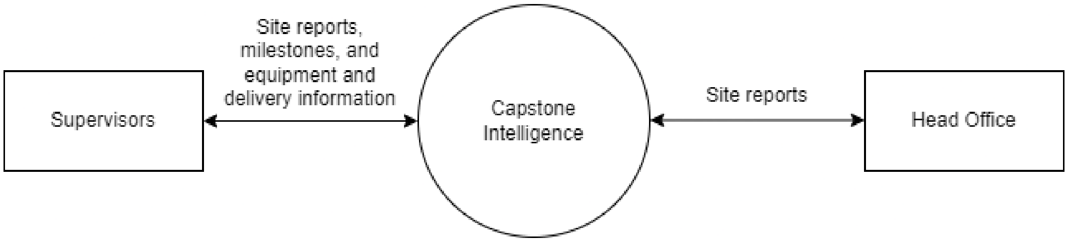
The system is event driven and thus requires the user to initiate operations from their side before the application responds. The application will accept information about the job site, including: progress on milestones, updates on deliveries and overall site reports. The application will store this data given by the user. When requested by the user, the application will give the user the information about the job site they requested. The application will remain in standby mode when not in use, and will refrain from making changes to data while in this mode.

Component Name	Component Purpose
Site Reporting	Manage and track all site reports for a jobsite
Milestone Tracking	Manage and track all project milestones
Site Information Tracking	Track all information specific to a jobsite
Delivery Scheduling	Track and manage all deliveries
Authentication	Manage app logins, logouts, and password changes
Database	Store all data

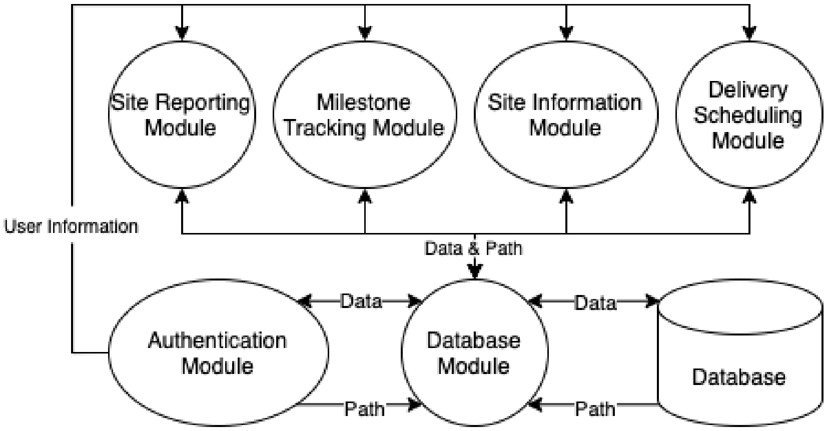
4.3 Undesired Event Handling

When an unexpected event occurs, the application should ensure that it enters a safe state as soon as possible. Entering a safe state will ensure that any improper or erroneous data is not accepted by the system. This will prevent further errors when a user attempts to read or modify corrupted or incorrect data. Preventing incorrect data from entering the system may require users to re-input their data in the case that an unexpected event occurs.

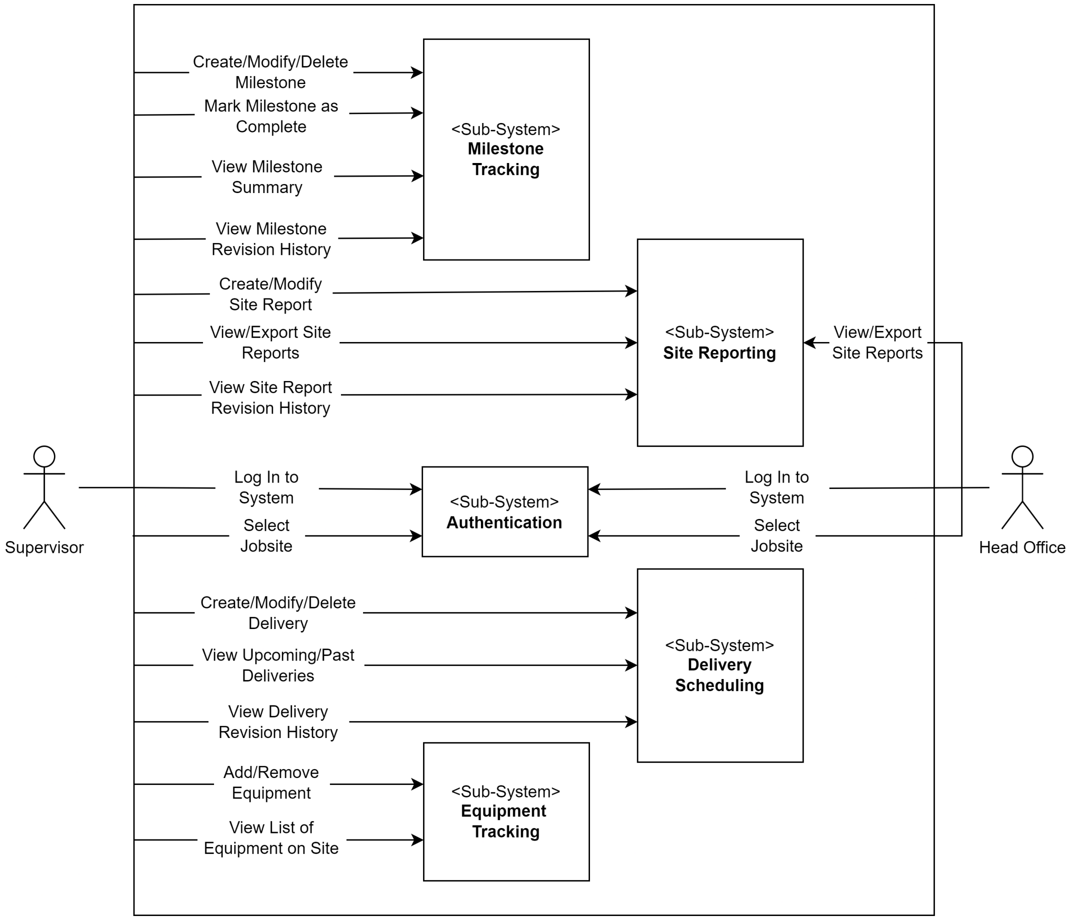
4.4 System Context Diagram


Site reports, milestones, and equipment and delivery information
Supervisors
Capstone Intelligence
Site reports
Head Office

4.5 Component Diagram



4.6 Functional Decomposition Diagram


Create/Modify/Delete Milestone
Mark Milestone as Complete
View Milestone Summary
View Milestone Revision History
<Sub-System> Milestone Tracking
Create/Modify Site Report
View/Export Site Reports
View Site Report Revision History
<Sub-System> Site Reporting
View/Export Site Reports
Supervisor
Log In to System
Select Jobsite
<Sub-System> Authentication
Log In to System
Select Jobsite
Head Office
Create/Modify/Delete Delivery
View Upcoming/Past Deliveries
View Delivery Revision History
<Sub-System> Delivery Scheduling
Add/Remove Equipment
View List of Equipment on Site
<Sub-System> Equipment Tracking

5 System Variables

5.1 Monitored Variables

Monitor Name	Monitor Type	Description
`user`	Object	Contains the current user's name, email address, assigned jobsites, current jobsite, and permission level

5.2 Controlled Variables

There are no controlled variables for this system.

5.3 Constants

There are no constants for this system.

6 Module Traceability

Module	Requirements
Authentication Module	AR1-3, PRR1-2
Site Information Module	JIR1-9, DSR5-6, DBR1, CCR1
Site Reporting Module	SR1-SR8, DBR1, PAR1, CPR2, IR6
Milestone Tracking Module	MTR1-MTR7, DBR1, IR6
Equipment Tracking Module	ETR1-ETR5, DSR5, DSR7, DBR1
Delivery Scheduling Module	DSR1-DSR8, DBR1, PAR1, IR6
Database Module	DBR1, CCR2, RER1, IR1, IR3, IR6

7 Custom Data Types

The application design uses many standard data types, such as strings. It also uses many custom-defined data types, which are defined below. These data types are all collections of key-value pairs.

7.1 Task

Key Name	Value Type	Description
`id`	String	A unique task identifier
`name`	String	The name of the task
`location`	String	The location of the task

7.2 WorkerGroup

Key Name	Value Type	Description
`trade`	String	The trade of the workers (eg. plumbing, electrical) present on the given jobsite on the given day
`numWorkers`	int	The number of workers present on the given jobsite on the given day

7.3 Subcontractor

Key Name	Value Type	Description
`id`	String	Unique subcontractor identifier
`name`	String	The name of the subcontractor

7.4 DailySub

Key Name	Value Type	Description
`subcontractor`	Subcontractor	The subcontractor whose information is being captured
`startTime`	Time	The time the subcontractor started working on a given day
`endTime`	Time	The time the subcontractor finished working on a given day
`location`	String	The location this subcontractor is operating on the jobsite.
`trades`	List⟨WorkerGroup⟩	The trades of workers a subcontractor has on the jobsite for a given day and how many of each were present
`tasks`	List⟨Task⟩	All tasks performed by a subcontractor on a given day
`comments`	String	Comments about a subcontractor for a given day

7.5 Postponement

Key Name	Value Type	Description
`date`	Date	The date the milestone was postponed to
`reason`	String	Reason why the milestone was postponed

7.6 User

Key Name	Value Type	Description
`name`	String	The name of the user
`email`	String	The email address associated with the user
`projects`	List⟨String⟩	The jobsite(s) the user has access to
`selectedSite`	String	The jobsite the user currently has selected
`admin`	Boolean	Whether or not the user is an admin user

7.7 Log

Key Name	Value Type	Description
`date`	Date	The date and time it was edited
`editedBy`	String	The user who made the modification
`fieldNameChanged`	String	The field name that was changed
`oldValue`	String	The old value of the field
`newValue`	String	The new value of the field

7.8 Milestone

Key Name	Value Type	Description
`id`	String	Unique milestone identifier
`name`	String	The name of the milestone
`createdBy`	String	The name of the person who created the milestone
`date`	Date	The current expected due date of milestone
`initialDate`	Date	The initial due date of the milestone
`completed`	Boolean	Whether the milestone has been completed or not
`subcontractors`	List⟨Subcontractor⟩	A list of subcontractors associated with this milestone
`postponements`	List⟨Postponement⟩	A list of all milestone postponements
`comments`	String	Comments associated with the milestone
`log`	Log	The revision log for the milestone

7.9 Delivery

Key Name	Value Type	Description
`id`	String	Unique delivery identifier
`name`	String	Name of delivery
`createdBy`	String	The name of the person who created the milestone
`date`	Date	Date of delivery
`startTime`	Time	Delivery start time
`endTime`	Time	Delivery end time
`location`	String	Delivery location
`subcontractor`	Subcontractor	Subcontractor assigned to the delivery (optional)
`equipment`	Equipment	Equipment required to unload the delivery (optional)
`log`	Log	The revision log for the delivery

7.10 Report

Key Name	Value Type	Description
`id`	String	A unique site report identifier
`date`	Date	The date the site report is created
`weather`	String	Weather information for the day
`subcontractors`	List⟨DailySub⟩	The information associated with the subcontractors present at the jobsite for the given day
`attendance`	List⟨String⟩	The names of all general contractors on site
`comments`	String	Comments associated with the site report
`images`	List⟨Object⟩	Photos attached to the site report
`editDateTime`	DateTime	Date and time the site report was created or modified
`currentlyEditing`	String	The username of the user currently editing the site report
`log`	Log	The revision log for the delivery

7.11 Equipment

Key Name	Value Type	Description
`id`	String	A unique equipment identifier
`name`	String	The display name of a piece of equipment
`color`	String	The display color of a piece of equipment
`deleted`	Boolean	Whether the equipment has been deleted or not

8 Module Guide

Below is a chart showing how the modules were decomposed.

Id	Name	Responsibility	Secret
1	App	Entire system	All secrets
1.1	Site Information	Manage (add, modify, and delete) and view all delivery locations, subcontractors, and employees of a given jobsite	The delivery locations, subcontractors, and employees for each jobsite
1.2	Site Reporting	Manage (add and modify) and view site reports of a specific jobsite	The site reports for each jobsite
1.3	Milestone Tracking	Manage (add, modify, and delete) and view milestones of a specific jobsite	The milestones for each jobsite
1.4	Equipment Tracking	Manage (add, modify, and delete) and view all equipment on a given jobsite	The equipment tracked at each jobsite
1.5	Delivery Scheduling	Manage (add, modify, and delete) and view deliveries of a specific jobsite	The deliveries for each jobsite
1.6	Authentication	Authenticates the user, allowing them to log in	User authentication
1.7	Database	Handles the read and write operations to the database	The database API authentication key

8.1 Site Information Module

8.1.1 Inputs

Input Name	Input Type	Description
`id`	String	A unique identifier
`name`	String	The display name of a new entry
`deleted`	Boolean	Whether the element has been deleted

8.1.2 Outputs

Output Name	Output Type	Description
`deliveryLocations`	List⟨DeliveryLocation⟩	All delivery locations at the current jobsite
`subcontractors`	List⟨String⟩	List of subcontractor id's of all subcontractors present on site jobsite
`employees`	List⟨String⟩	A list of employee emails (as Strings) of all employees at the current jobsite

8.1.3 Description of Behaviour

For `deliveryLocations`, this module will add/remove a delivery location (uniquely identified by `id`) with the name `name` to/from the current jobsite. For `subcontractors` and `employees`, this module will add/remove an element (uniquely identified by `id` or `email`) with the name `name` to/from the current jobsite. It will also provide a list of each, as seen above.

This functionality will be achieved by accessing (and modifying, if applicable) the relevant database. The value of `id` will be generated through an external library upon creation and `name` will be entered by the user.

8.1.4 Exception Handling

Input validation will be the main form of exception handling. All mandatory values will be checked to ensure they are not empty, and will prevent the user from saving their edits while these empty mandatory values exist. Exceptions will be caught when accessing data from the database in case of invalid data in the database.

8.1.5 Timing Constraints

N/A

8.1.6 Initialization

Upon navigation to either the delivery locations, subcontractors, or employees screen, the user will see all elements currently tracked at the current jobsite (if any). If the user is an admin, they will also see a button to add an element to the current jobsite and a button to delete each element.

8.2 Site Reporting Module

8.2.1 Inputs

Input Name	Input Type	Description
`id`	String	A unique site report identifier
`date`	Date	The date the site report is created
`weather`	String	Weather information for the day
`subcontractors`	List⟨DailySub⟩	All subcontractors and their daily information associated with the site report
`attendance`	List⟨String⟩	The names of all general contractors on site
`comments`	String	Comments associated with the site report
`images`	List⟨Object⟩	Photos attached to the site report
`editDateTime`	DateTime	Date and time the site report was created or modified
`currentlyEditing`	String	The username of the user currently editing the site report
`username`	String	The current user's username

8.2.2 Outputs

Output Name	Output Type	Description
logs	List⟨Log⟩	A list of revision logs for the site report
report	Report	The site report on a given date
pdf	PDF	A PDF version of a site report for a given date to be used by head offices

8.2.3 Description of Behaviour

This module will handle the creation and modification of site reports (selected via date and uniquely identified by id). Each site report will contain the fields weather, subcontractors, attendance, comments, and images. These fields in the site report are empty by default and will be optional for users to input. The Subcontractor objects contained in subcontractors will contain the inputs id, subName, startTime, endTime, numWorkers, classWorkers, location, tasks, and comments. Each subcontractor's startTime and endTime will be accurate to the minute. The fields weather, comments, location, tasks and trades are optional for the subcontractor information. startTime and endTime are given default values based on the standard eight-hour work day. These fields will be used to create the outputs report and pdf. The inputs username, editDateTime and the output report will be used to create the output logs.

This functionality will be achieved by accessing (and modifying, if applicable) the relevant database. The value of a subcontractor's id will be automatically generated when a subcontractor is added to a site report for the first time. editDateTime will be automatically created using the Date and time the user created or modified the site report, and currentlyEditing will be automatically populated with username for as long as the user is editing the report. The value of id will be generated through an external library upon creation, username will be taken from the user system variable, and the rest of the fields will be entered by the user.

8.2.4 Exception Handling

Input validation will be the main form of exception handling. All mandatory values will be checked to ensure they are not empty, and will prevent the user from saving their edits while these empty mandatory values exist. Additional input validation will be implemented to ensure that each endTime associated with each subcontractor does not occur after its startTime and that each numWorkers is an integer. Exceptions will be caught when accessing data from the database in case of invalid data in the database.

8.2.5 Timing Constraints

N/A

8.2.6 Initialization

Upon the first navigation to the site reporting screen, the user will first see a calendar view where they can select a Date to create a new site report. After selecting a date, the user will be presented with the buttons allowing them to input data related to general contractors as well as a button to add new subcontractors to the report.

8.3 Milestone Tracking Module

8.3.1 Inputs

Input Name	Input Type	Description
`id`	String	A unique milestone identifier
`name`	String	The name of the milestone
`date`	Date	The current expected due date of milestone
`initialDate`	Date	The expected due date of the milestone at its creation time
`completed`	Boolean	Whether the milestone has been completed or not
`subcontractors`	List⟨Subcontractor⟩	All subcontractors associated with this milestone
`postponements`	List⟨Postponement⟩	All postponements of the current milestone
`comments`	String	Comments associated with the milestone
`username`	String	The current user's username

8.3.2 Outputs

Output Name	Output Type	Description
`logs`	List⟨Log⟩	All revision logs for the associated milestone
`milestones`	List⟨Milestone⟩	All milestones for a given date
`tasks`	List⟨Task⟩	All tasks associated with any milestone

8.3.3 Description of Behaviour

This module will create/delete/modify a milestone (selected via `name` and uniquely identified by `id`) to/from the current jobsite. It will also provide a list of all milestones on a specified `date` at a specified `jobsite`, a list of revision logs for a specified milestone (selected via `name` and uniquely identified by `id`), and a list of tasks on a specified `date` at a specified `jobsite`. The list of tasks associated with a milestone will be based on the tasks completed by the subcontractors specified in `subcontractors`. The value of `logs` will be generated by the system after every modification to a milestone using the `username`, the field that was edited, the old value, the new value, and the time of edit.

This functionality will be achieved by accessing (and modifying, if applicable) the relevant database. The value of `id` will be generated through an external library upon creation, `username` will be taken from the `user` system variable, and the rest of the fields will be entered by the user.

8.3.4 Timing Constraints

N/A

8.3.5 Exception Handling

Input validation will be the main form of exception handling. All mandatory values will be checked to ensure they are not empty, and will prevent the user from saving their edits while these empty mandatory values exist. Exceptions will be caught when accessing data from the database in case of invalid data in the database.

8.3.6 Initialization

Upon navigation to the milestone screen, the user will see all the milestones corresponding to the current jobsite, and all the tasks corresponding to the current job site and associated with any of the milestones. There will be a button to create a new milestones for the jobsite, along with the option to modify and delete any of the milestones.

8.4 Equipment Tracking Module

8.4.1 Inputs

Input Name	Input Type	Description
`id`	String	A unique equipment identifier
`name`	String	The display name of a piece of equipment
`color`	String	The color name of a piece of equipment
`deleted`	Boolean	Whether the equipment has been deleted or not

8.4.2 Outputs

Output Name	Output Type	Description
`equipment`	List⟨Equipment⟩	All equipment being tracked at the current jobsite

8.4.3 Description of Behaviour

This module is responsible for adding, modifying, or removing a piece of equipment (uniquely identified by `id`) with the name `name` and color `color` associated with the current jobsite. It will also provide `equipment`, the list of all equipment present at a specified jobsite.

This functionality will be achieved by accessing (and modifying, if applicable) the relevant database. The value of `id` will be generated through an external library upon creation and `name` and `color` will be entered by the user.

8.4.4 Exception Handling

Input validation will be the main form of exception handling. All mandatory values will be checked to ensure they are not empty, and will prevent the user from saving their edits while these empty mandatory values exist. Exceptions will be caught when accessing data from the database in case of invalid data in the database.

8.4.5 Timing Constraints

N/A

8.4.6 Initialization

Upon navigation to the equipment tracking screen, the user will see all equipment currently tracked at the current jobsite (if any). If the user is an admin, they will also see a button to add a piece of equipment to the current jobsite and a button to delete each piece of equipment.

8.5 Delivery Scheduling Module

8.5.1 Inputs

Input Name	Input Type	Description
`id`	String	A unique delivery identifier
`name`	String	The name of the delivery
`date`	Date	The date of the delivery
`startTime`	Time	The delivery's start time
`endTime`	Time	The delivery's end time
`location`	String	The delivery's location on site
`subcontractor`	Subcontractor	The subcontractor assigned to the delivery (optional)
`equipment`	Equipment	The equipment required to unload the delivery (optional)
`username`	String	The current user's username

8.5.2 Outputs

Output Name	Output Type	Description
`deliveries`	List⟨Delivery⟩	The deliveries for the current jobsite for a given date
`logs`	List⟨Log⟩	All revision logs for a given delivery

8.5.3 Description of Behaviour

This module is responsible for adding, modifying, and removing a delivery (uniquely identified by `id`) with the fields `name`, `date`, `startTime`, `endTime`, `location` and the two optional fields `subcontractor` (the subcontractor assigned to the delivery) and `equipment` (the equipment required to unload the delivery) associated with the current jobsite. The values of `startTime` and `endTime` will be accurate to the minute. It will also provide `deliveries`, the list of deliveries for a given `jobsite` on a specific `date`, and `logs`, a list of all revision logs, each generated after every modification to a milestone using from `username`, the edited field, its old value, its new value, and the time of edit.

This functionality will be achieved by accessing (and modifying, if applicable) the relevant database. The value of `id` will be generated through an external library upon creation, `username` will be taken from the `user` system variable, and the rest of the fields will be entered by the user.

8.5.4 Exception Handling

Input validation will be the main form of exception handling. All mandatory values will be checked to ensure they are not empty, and will prevent the user from saving their edits while these empty mandatory values exist. Additional input validation will be implemented to ensure that `endTime` does not occur after `startTime`. Exceptions will be caught when accessing data from the database in case of invalid data in the database.

8.5.5 Timing Constraints

N/A

8.5.6 Initialization

Initially, the user will see an empty “day view”, similar to Google Calendar’s “day view”, with an option to add a delivery to that day.

8.6 Authentication Module

8.6.1 Inputs

Input Name	Input Type	Description
`email`	String	The user’s email which acts as its username
`password`	String	The user’s password

8.6.2 Outputs

Output Name	Output Type	Description
`user`	User	The authorized user’s information

8.6.3 Description of Behaviour

This module will use the `email` and `password` to authenticate the user and retrieve all their information, `user`, which will be required to view the jobsite’s data.

8.6.4 Exception Handling

The heavy lifting for authentication is handled by the Firebase SDK, which means only basic authentication functionality needs to be implemented.

However, there are still exceptions caught which are detailed in the chart below. They are all of type `FirebaseAuthException` and are used when signing in and creating an account.

Exception Type	Caught Exception
`user-not-found`	No user found for the supplied email.
`wrong-password`	Wrong password provided for that user.
`weak-password`	The password provided is too weak.
`email-already-in-use`	An account already exists for that email.

8.6.5 Timing Constraints

N/A

8.6.6 Initialization

The user will initially be met with a login screen where they will need to enter their login credentials to access the jobsite's data.

8.7 Database Module

8.7.1 Inputs

Input Name	Input Type	Description
`data`	Object	The new, or edited data to be added to the database
`date`	DateTime	The date of the document where the data is stored.
`collectionType`	String	The type of collection (e.g. Milestones, Deliveries, Equipment) where the data is stored.
`id`	String	The id of the object to either be edited or deleted.

8.7.2 Outputs

Output Name	Output Type	Description
`data`	Object	The data returned from the database

8.7.3 Description of Behaviour

This module will use the `date` and `collectionType` (formatted to work with the database used, Firebase Cloud Firestore) to be able to find the stored `data` at that location and return it. It will also be able to use the `date` and `collectionType` and `data` as inputs to store the given `data`. Finally, it will use the `date`. `collectionType`, `id`, and `data` as inputs to store the edited `data`

8.7.4 Exception Handling

There are two primary methods of handling exception for this module. The first occurs when returning data. The snapshot taken from the database is checked to see if it is null; if so, a generic `Exception` with an error message is thrown. The second occurs when inputting data. Since input validation is the main form of exception handling and is performed when the user provides input to the system through other modules, there is no need to implement any additional exception handling for the database module.

8.7.5 Timing Constraints

N/A

8.7.6 Initialization

N/A