Spatial Crowdsourcing for Dynamic Settings

SDMay22-31

Goce Trajcevski	-	Client/Faculty Advisor
Abdula Eljaam	-	Testing
Aman Agarwal	-	Individual Component Design
Cole Dulaney	-	Team Organization
Isaac Reed	-	Team Organization
Seth Platt	-	Client Interaction
Shagun Bansal	-	Individual Component Design
Yuichi Hamamoto	-	Testing

Email - sdmay22-31@iastate.edu
Website - https://sdmay22-31.sd.ece.iastate.edu/

Executive Summary

Development Standards & Practices Used

- Agile Task Management Methodology
- Object-Oriented Programming
- Data-Structures
- Algorithms
- IEEE Std 1228 - Standard for Software Safety Plans
- IEEE Std 1012 - Standard for Software Verification and Validation
- IEEE Std 1219 - Standard for Software Maintenance
- IEEE Std 1063 - Standard for Software User Documentation
- IEEE Std 982.1 - Standard of Measures to Produce Reliable Software
- IEEE Std 1008-1987 - Standard for Software Unit Testing
- IEEE 12207.2-1997 - Industry Implementation of International Standard ISO/IEC 12207 : 1995 (ISO/IEC 12207) Standard for Information Technology - Software Life Cycle Processes - Implementation Considerations.

Summary of Requirements

- Research Spatial Crowdsourcing Algorithms
- Understanding Firebase for Use as a Database and Host Server and Applying that Knowledge
- Developing Mobile and Web-Based Applications to Allow for Utilization of the Algorithm

Applicable Courses from Iowa State University Curriculum

- COM S 230: Discrete Computational Structures
- COM S 227: Object-Oriented Programming
- COM S 228: Introduction to Data Structures
- COM S 309: Software Development Practices
- COM S 311: Introduction to the Design and Analysis of Algorithms
- COM S 363: Introduction to Database Management Systems
- CPR E 310: Theoretical Foundations of Computer Engineering
- SE 329: Software Project Management
- SE 339: Software Architecture and Design
- SE 409: Software Requirements Engineering

New Skills/Knowledge acquired that was not taught in courses

- Firebase
- GoogleMaps API
- ReactJS
- Algorithms for Task Assignment Based on Worker’s SpatioTemporal Data

Tables and Figures

Tables

Figures

Table of Contents

1 Team

Details regarding SDMAY-31 and its members

1.1 Team Members

1. Abdula Eljaam
2. Aman Agarwal
3. Isaac Reed
4. Seth Platt
5. Cole Dulaney
6. Shagun Bansal
7. Yuichi Hamamoto

1.2 Required Skill Sets

1. Familiarity with SCRUM / Agile
2. Web and Mobile application development
3. Database management

1.3 Skill Sets Covered by Team

- **Javascript, HTML, CSS:** Cole Dulaney, Aman Agarwal, Isaac Reed
- **Web application development:** Cole Dulaney, Aman Agarwal, Isaac Reed, Shagun Bansal
- **Mobile Application Development:** Seth Platt, Aman Agarwal, Yuichi Hamamoto
- **Database management:** Isaac Reed, Yuichi Hamamoto

1.4 Project Management Style

For our project, our team will adopt the agile methodology. This style of management is extremely common both in industry as well as in software development projects like ours'.

1.5 Initial Project Management Roles

1.	Cole Dulaney	-	Team Organization
2.	Isaac Reed	-	Team Organization
3.	Yuichi Hamamoto	-	Testing
4.	Aman Agarwal	-	Individual Component Design
5.	Shagun Bansal	-	Individual Component Design
6.	Seth Platt	-	Client Interaction
7.	Abdula Eljaam	-	Testing

2 Introduction

This semester our group, SDMAY-31, was tasked with creating both an Android and Web application to solve the problem of applied dynamic spatial crowdsourcing.

2.1 Problem Statement

The issue of spatial crowdsourcing involves three main components: tasks, workers, and a platform. However, In **dynamic** spatial crowdsourcing, the workers and clients have a location associated with them. In this case, workers are assigned tasks by the platform in real-time based on various possible optimization methods. These methods may include maximizing total number of assigned tasks, total payoff of tasks assigned to workers, and minimization of total time spent and distance traveled by workers.

The main objective of this project is to develop a system that will implement algorithmic solutions to the settings of dynamic spatial crowdsourcing. Specifically, our system will attempt to optimize the number of tasks completed within certain time limits. Time-permitting we may consider different optimization criteria such as minimizing the time for completing a given set of tasks. Our system will target two kinds of users - Workers who execute tasks and Clients who post requests for tasks.

2.2 Requirements & Constraints

Functional requirements:

- The android and web app should be able to display status of assignment and worker in real-time
- Only appropriate users should be able to see the location/info of workers
- Interpret existing worker data to decide optimal routes and assignments
- The applications should be scalable for multiple users
- Job requests are updated in real time
- Workers must be able to post their availability and location
- Application should calculate the optimal assignment of worker to task
- Users should be able to view status of their requests in real-time
- Should account for removal of available jobs or workers, and adjust assignments accordingly
- Proper authentication for user accounts
- Data storage and tracking of completed tasks
- Optimized for fast response time

Non-functional requirements:

- Accessible from mobile devices and PC
- Design should be extensible.
- Data should be stored safely and accurately, and its integrity shall be maintained across API hits

- Clean and best practice code to improve future maintainability
- Application should be functioning reliably
- Privacy and security of workers
- Intuitive and “clean” web and mobile UI

Constraints:

- Budget - for computational resources (Server)
- Time [1250 hours for two semesters]
- Updates should be propagated to users within 3 to 5 seconds.
- Must abide by all privacy and business laws regarding all users’ data
- Visualization of location on the map should be accurate.

2.3 Engineering Standards

The following table lists IEEE standards we shall abide by, and why they are relevant to our project’s success.

Engineering Standards	Justification
IEEE Std 1228, Standard for Software Safety Plans	As we will require the usage/storage of sensitive information, we must take security seriously and plan around it.
IEEE Std 1012, Standard for Software Verification and Validation	We must ensure that our implementation meets the client’s requirements, and does everything they need it to do. I.e we must completely address the problem statement.
IEEE Std 1219, Standard for Software Maintenance	We must safely and actively maintain the application to provide the best possible user experience. Data integrity and security must be maintained throughout this process as well.
IEEE Std 1063, Standard for Software User Documentation	Our application and service must be documented properly; both for users and future developers. This will impact scalability and maintainability, among many other factors.
IEEE Std 982.1, Standard Dictionary of Measures to Produce Reliable Software	As a company’s day-to-day operations may revolve around this application, we must provide a reliable service for them to use
IEEE 1008-1987 - IEEE Standard for Software Unit Testing	In order to ensure our application functions properly and meets reliability standards we

	will follow the accepted standards for software testing.
IEEE 12207.2-1997 - IEEE/EIA Guide - Industry Implementation of International Standard ISO/IEC 12207 : 1995 (ISO/IEC 12207) Standard for Information Technology- Software Life Cycle Processes - Implementation considerations	We will follow industry standards and best practices throughout our development process both to ensure a quality final product, and to create a well organized development process. This can be done via the team's Agile based development methodology.

Table 1: Applicable Engineering standards

2.4 Intended Users and Uses

Users / Actors:

- Worker
- Client/Requester
- Company Administrator
- Application Administrator

Use Case	Applicable Roles / Actors	Details
New User Registration	All	Like any other app, users of all roles must be able to create an account on our application.
Task assignment similar to services such as GrubHub	Worker, Admin	The system will analyze the data and make the best possible task assignment. Workers and Admins may be able to manually take jobs in some cases.
View status of assignments/workers in real-time	Client, Admin	Authorized users should be able to view the status of a job as the worker executes it.
Live location tracking of workers completing tasks	Admin	Admins, with appropriate permissions, can see the location of a driver while they are executing a task.
Individual worker task stats displayed	Admin, worker	An admin or worker can view the statistics for said worker's

		task history.
Claim task completion	Worker	When completing the job the worker submits a claim to the app that they completed it. This process is similar to popular modern delivery services, where the driver must take a picture of the delivery to prove it was completed.
Confirm task completion	Client	The client can confirm or reject the completion of the job. If they reject it, they must provide a reason. Either choice will notify the worker and the boss.
Remove/add workers	Admin	An admin should be able to remove or add workers from/to the list of available workers in the system.
Remove/add tasks	Client	A client should be able to remove or add tasks from/to the list of available tasks in the system.
Worker clock in/out	Worker	The worker should be able to clock in and out of work, or at least the service, via the app.
Setting worker status and personal settings	Worker	Workers should be able to set their availability, adjust their list of skills, personal info, etc.

Table 2: Breakdown of Use Cases and General Functionality Descriptions

3 Project Plan

Detailed plan for solving the problem of Dynamic Spatial Crowdsourcing via our application

3.1 Project Management & Tracking Procedures

We will be using the agile project management style for our project. Agile is very commonly used in software, especially for both web and mobile application projects, which are also two of the deliverables for our project. Our goals and objectives will also likely be small, and incremental, which is another characteristic of agile project management.

We will use Gitlab to track version history and to allow for simultaneous development. Git issue boards will be used to organize and assign tasks. Alongside this, Gitlab milestones and epics will assist in tracking progress, and ensure the development process is moving forward in a timely manner. Group communication, useful resources, and meeting minutes will be organized in the team discord server. Important documents and reports will be stored in a shared Google Drive folder.

3.2 Task Decomposition

The tasks necessary to implement our solution to Dynamic Spatial Crowdsourcing are outlined as follows:

Task 1 - Finish studying background knowledge on Spatial Crowdsourcing

a. Familiarize ourselves with the specifics of the project and problem statement
b. Background research on similar projects to find best practices

Task 2 - Complete requirements engineering process and planning.

a. Identify relevant IEEE standards that must be followed
b. Recognize users and build use-cases for project

Task 3 - Complete review of client-provided studies and scope definition of project.
Task 4 - Document and present original project plan for client approval
Task 5 - Solidify design ideas and concepts for both UI and structural implementation.
Task 6 - Finalize selection of application platforms, frameworks, languages, and tools.
Task 7 - Complete documentation for the final design proposal, testing plans, and development/implementation plan.
Task 8 - Specifics subject to change with time and client request.

a. Implementation of the applications
b. Revisions based on client feedback

Task 9 - Final application testing should be completed.

a. Final pre-launch preparations
b. Presentation of application and appropriate documentation to stakeholders.
c. Deployment of final product

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

The project will be evaluated with the use of standard agile charts and documentation. The development team and stakeholders may also use our non-functional requirements, as described in the design document, for evaluation. The specific criteria on which we will be evaluating the project's progress is as follows:

1. User experience elements such as response time across all platforms
2. Design should be extensible
3. Data should be stored safely so that only authorized users may access it
4. Worker location data should be stored accurately in terms of proximity to exact location as best attainable by GPS sensors
5. Clean and best practice code to improve future maintainability
6. Application should function as expected under heavy load
7. Privacy and security of workers should be maintained
8. The UI for both web and mobile should be Intuitive and appealing to the users

These metrics will be organized and evaluated using the following tools:

1. Burndown Chart
2. Gantt Chart
3. Work Breakdown Structure (WBS)
4. Sprint Reports
5. Code reviews by peers
6. Web server load testing
7. JMeter (API testing tool)

3.3.1 Fall Semester 2021 (491) Milestones

September

1. Familiarize ourselves with the specifics of the project and problem statement
2. Completed first stage of requirements engineering process

October

1. Application scope and industry finalized
2. Project plan document finalized and accepted by client

November

1. Solidified design ideas for both UI and structural implementation.
2. Selection of application platforms, frameworks, languages, and tools are finalized and documented

December

1. Documentation for the final design proposal, testing plans, and development/implementation plan completed

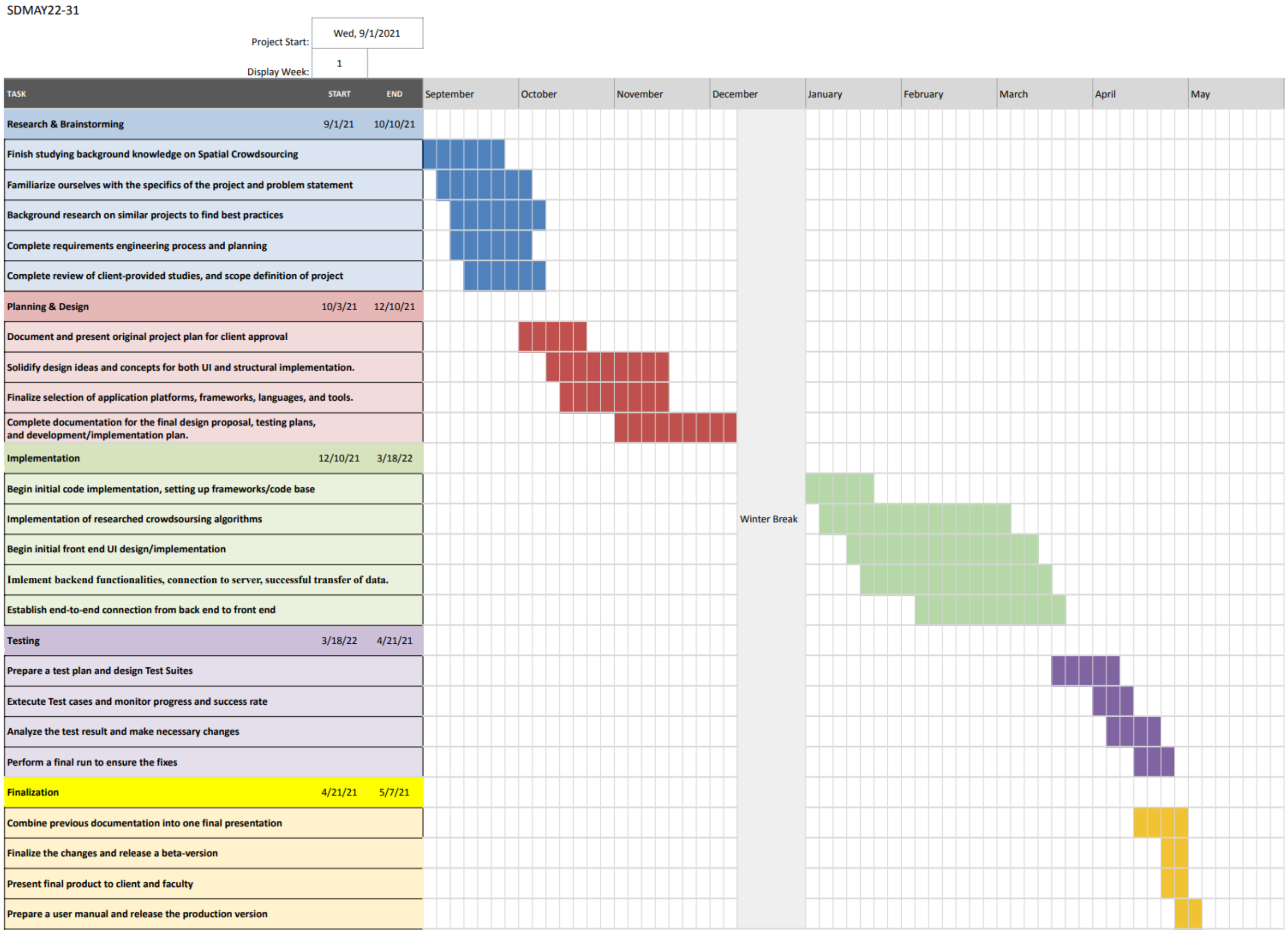
3.3.2 Spring Semester (492) Milestones

These are the proposed milestones, exact dates will be determined during 492.

1. Worker location data controller optimized such that the data is reflected in real time. This means it shall stay within the standard margin of **1-10 seconds** to update.
2. Data controllers optimized such that all important changes should be disclosed to the relevant users in the same **1-10 second** real time window. An example of these changes would be the addition/removal of a task or driver.
3. The worker assignment algorithm completed, and optimized for minimal runtime, such that it fulfills the proposed quality attribute. Clients now receive a timely UI response confirming this process has occurred.
4. The application can now smoothly support ***X*** users at once. (**X** being subject to capabilities of server provided by client)
5. The application's functionalities have all been optimized and connected smoothly. It now responds within the standard **1 second** time frame. If a request takes longer, some indication is provided to ensure the user something is happening and the app has not stopped.

3.4 Project Timeline

With respect to the information described in 3.1 - 3.3, Figure 1 represents the project's timeline timeline



SDMAY22-31

Project Start: Wed, 9/1/2021

Display Week: 1

TASK	START	END
Research & Brainstorming	9/1/21	10/10/21
Finish studying background knowledge on Spatial Crowdsourcing		
Familiarize ourselves with the specifics of the project and problem statement		
Background research on similar projects to find best practices		
Complete requirements engineering process and planning		
Complete review of client-provided studies, and scope definition of project		
Planning & Design	10/3/21	12/10/21
Document and present original project plan for client approval		
Solidify design ideas and concepts for both UI and structural implementation.		
Finalize selection of application platforms, frameworks, languages, and tools.		
Complete documentation for the final design proposal, testing plans, and development/implementation plan.		
Implementation	12/10/21	3/18/22
Begin initial code implementation, setting up frameworks/code base		
Implementation of researched crowdsoursing algorithms		
Begin initial front end UI design/implementation		
Imlement backend functionalities, connection to server, successful transfer of data.		
Establish end-to-end connection from back end to front end		
Testing	3/18/22	4/21/21
Prepare a test plan and design Test Suites		
Extecute Test cases and monitor progress and success rate		
Analyze the test result and make necessary changes		
Perform a final run to ensure the fixes		
Finalization	4/21/21	5/7/21
Combine previous documentation into one final presentation		
Finalize the changes and release a beta-version		
Present final product to client and faculty		
Prepare a user manual and release the production version		

Figure 1: Gantt Chart

3.5 Risks and Risk Management

The following is the table we designed to describe any risks we may face, and how we intend on mitigating them.

* Rank = Relative Severity * Relative Probability

Relative Severity	Relative Probability	Phase	Description	Mitigation	Rank*
8	8	Design	Client base size unknown	Conduct the market size research and expand the existing database accordingly.	8
1	6	Design/ Coding	Requirement not well defined cause wasting time on unnecessary features	Consult the decided use cases. Consultation with faculty. Adopt Agile refinements.	2
7	7	Design/ Coding	Task members don't like doing designated tasks	Negotiate optimal delegation of tasks	7
2	5	Design/ Coding	Task members are falling behind in their assigned task	Regularly check the progress of team members to ensure help on any difficulties faced	3
3	1	Coding	Developer inexperience with the required project technologies	Have dedicated learning time during work week and sharing knowledge between team members	1
4	4	Coding	Change in requirements aren't compatible with existing architecture	Implement a flexible and extensible architecture.	5
6	3	Training	Clients/Workers are unfamiliar with software	Create user-friendly tutorials on the app.	6
5	2	Testing	Tests not covering all algorithmic behaviours.	Writing quality tests. Using visualization tools for covering all	3

				program paths.	

Table 3: Risk Rankings

Risk Based

Firing Event	Probability	Cost in 1000s
Client base size unknown	0.2	1
Requirement not well defined cause wasting time on unnecessary features	0.6	10
Clients/Workers are unfamiliar with software	0.75	3
Developer inexperience with the required project technologies	0.8	7
Tests not covering all algorithmic behaviours.	0.75	5
Task members don’t like doing designated tasks	0.4	2
Change in requirements aren’t compatible with existing architecture	0.7	6
Task members are falling behind in their assigned task	0.7	8

Table 4: Risk Cost

To get an intuitive idea of what the impact of this risk assessment is, here is an approximate value:
a = 1.2 * 9 *0.0043 + 3.5
= 3.55
≈ 4
Cost for the top 4 risks is $31,000, which will be set aside as contingency funds.
This value is estimated based on a real-world large scale deployment. In the context of this Senior Design project, we don’t actually need to deal with these risk costs.

3.6 Personnel Effort Requirements

For each task, we are estimating the completion time considering all seven team members spending an average of half an hour a day per person.

Task	Estimated Completion Time (in hours)
Sept. 1 - Finish studying background knowledge on Spatial Crowdsourcing	7 workers, 7 days 7 * 6 = 42 hours
Sept 10 - Familiarize ourselves with the specifics of the project and problem statement	7 workers, 3 days 7 * 3 = 21 hours
Sept 15 - Background research on similar projects to find best practices	7 workers, 3 days 7 * 3 = 21 hours
Sept 25 - Complete requirements engineering process and planning.	7 workers, 8 days 7 * 8 = 56 hours
Oct. 1 - Complete review of client-provided studies, and scope definition of project.	7 workers, 7 days 7 * 6 = 42 hours
Oct 10 - Document and present original project plan for client approval	7 workers, 14 days 7 * 12 = 84 hours
Nov. 1 - Solidify design ideas and concepts for both UI and structural implementation.	7 workers, 10 days 7 * 9 = 63 hours
Nov. 15 - Finalize selection of application platforms, frameworks, languages, and tools.	7 workers, 8 days 7 * 7 = 49 hours
Dec. 1 - Complete documentation for the final design proposal, testing plans, and development/implementation plan.	7 workers, 10 days 7 *9 = 63 hours
Jan - May - Specifics subject to change with time and client request. Implementation of the applications	7 workers, 100 days 7 * 90 = 630 hours

May - Final application testing should be completed.	7 workers, 15 days 7 * 14 = 98 hours

Table 5: Estimated Time Spent

3.7 Other Resource Requirements

1. Remote web server to host the database and provide logic assignments, with one provided by the school no additional acquisition is required by this.
2. Physical devices for testing of both the web client and the android applications, applicable devices are already owned by stakeholders meaning no additional device acquisition should be required.

4 Design

Detailed explanation of the application's design process from conceptualization to finalization with regards to project requirements

4.1 Design Context

Broad description of the requirements, challenges, and context of the application

4.1.1 Broader Context

Our broad context is a situation where we have N job-sites, M workers, who each have their own skill descriptions and respective locations. As we are crowdsourcing our workers will be assigned to these job-sites based on various factors depending upon the work situation. We are targeting individual consumers and service providers by our model. This application will increase competition in our communities and thus provide better and more affordable services.

Below are a few areas and descriptions of how the project would have the ability to impact them.

Area	Description	Examples
Public health, safety, and welfare	Workers - 1. By increasing economic opportunity for workers in the community, this will also positively impact overall health and welfare, as workers will be able to more comfortably afford more of their expenses. 2. The workers who respond to the tasks may take the risk of responding to malicious users: i.e. users who "pretend" to have an issue, but make a job request for another (possibly malicious) reason. Mitigating the amount of travel workers do is also important, which would decrease the likelihood of a traffic accident or other travel-related issues. Users - 1. Users should see health/safety increase with the use of our solution, as users may request jobs that, if left uncompleted, could impact their	Workers - 1. A worker who previously struggled in being able to find jobs is given more opportunities (which results in increased mental wellbeing for workers) as a direct result of the algorithmic solution implemented. 2. A malicious user uses the app to request a job, but when the worker arrives at the job site, neither the user nor the job is anywhere to be found. Another applicable example would be requesting an Uber driver, but cancelling last-minute, resulting in wasted time and money for the worker. Users - 1. A user posts a job request for a leaky pipe which is quickly completed by a worker. If left unattended, this pipe could've caused harmful effects to the user's health. 2. A user makes a job request to fix a leaky pipe that is assigned (unintentionally) to a malicious or

	physical or mental health and welfare. 2. Similar to workers, users may take the risk of “malicious” workers. These would be workers who use the product and are in some way negligent to the job they are assigned. This would also have the same effect if a user posts a job they need done urgently, but the job is never completed by a worker.	negligent worker that doesn’t show up, or does not complete the job. This could result in not only damage to the user’s plumbing and infrastructure, but also health risk to the user due to mold if not attended to in a timely manner.
Global, cultural, and social	It will help organize professional services and their execution by professional workers and help with the organization and transportation of those with similar social desires.	Development or operation of the solution would violate a profession’s code of ethics, implementation of the solution would require an undesired change in community practices
Environmental	By dynamically optimizing routes of workers the emissions from workers vehicles will be reduced between jobs.	With algorithmically calculated optimized routes for the workers, excess car emission can be controlled helping the environment.
Economic	It enables more competition amongst service providers which in turn can provide more competitive pricing. More competition also promotes better service and enables more flexibility for the customers. The proposed solution would also create more jobs for workers in the community/area in which it’s implemented.	If a customer insists on having his job finished in a certain hour and is willing to pay more, the system should enable such assignments.

Table 6: Breakdown of Broader Context

4.1.2 User Needs

Workers need to be able to do the following:

1. Accept tasks to confirm for the users and crowdsourcing algorithm.
2. Get the location of the requested task to be able to arrive and accomplish the task.
3. Post their location, skills and task status because both clients and the crowdsourcing algorithm require this information.
4. Finish accepted tasks in a timely manner to ensure work done via the app is done properly.

5. Regularly monitor task schedules for any updates because of the dynamic nature of the problem.
6. File feedback/issues if any, so it can be handled in a timely manner.

Clients/Requesters need to be able to do the following:

1. be able to post tasks so there are tasks to calculate assignments for, and so the application can serve their needs.
2. be able see the status of assignment/workers in real time so they can stay up to date on the status of their task.
3. be able to confirm the task completion to ensure the application provided them proper services
4. be able to approve funding for the requested task so the algorithm can provide them with a worker in their price range, and so the worker can receive payment.
5. be able to file feedback/issues if any so it can be handled in a timely manner.

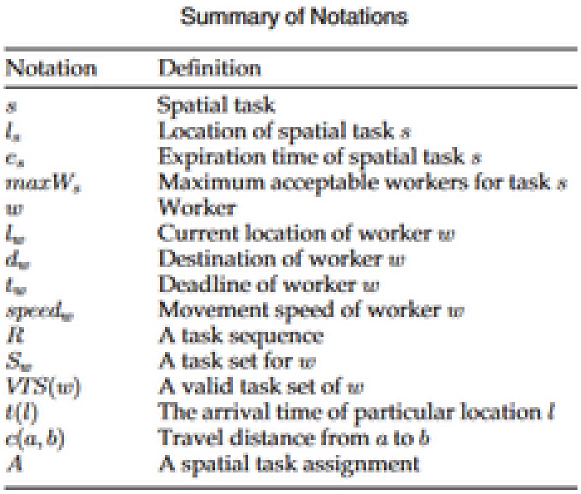
Administration needs to be able to do the following:

1. Assess new workers based on the given information to ensure proper sign up, and qualified workers.
2. Manage customer-worker conflicts and feedback so they can be handled in a timely manner.
3. Issue incentives to workers based on their performance so that the application can keep workers motivated, and draw in new users.
4. Build a customer appreciation strategy to boost business and user satisfaction.
5. Manage privacy protection to ensure all user data remains private.

4.1.3 Prior Work/Solutions

Detailed background information on the problem of dynamic spatial crowdsourcing, and programmatic solutions, can be found in the "*Destination-aware task assignment in spatial crowdsourcing: A worker decomposition approach*" report.

In the modern day, there are actually many applications which intend to solve the issue of spatial crowdsourcing for a specific subgroup of tasks. A few prominent examples are rideshare services such as Uber or Lyft, and food delivery services such as DoorDash and Eat Street. These companies provide similar tools, however they only handle location data; not worker skills / and differing tasks. Due to this, these popular solutions actually lack many of the criteria shown in figure 1, and cannot truly be considered examples of solutions to our problem.



Summary of Notations

Notation	Definition
s	Spatial task
l_s	Location of spatial task s
e_s	Expiration time of spatial task s
$maxW_s$	Maximum acceptable workers for task s
w	Worker
l_w	Current location of worker w
d_w	Destination of worker w
t_w	Deadline of worker w
$speed_w$	Movement speed of worker w
R	A task sequence
S_w	A task set for w
$VTS(w)$	A valid task set of w
$t(l)$	The arrival time of particular location l
$c(a, b)$	Travel distance from a to b
A	A spatial task assignment

Figure 2: Zhao, Y et. all (2019, June 12)

We may be able to draw information from these services such as location assignment and UI designs, but we will need to make substantial changes in order to provide a proper solution. With these changes in mind, the examples and information provided in the “*Destination-aware task assignment in spatial crowdsourcing”* report will likely prove to be valuable to our development process. This report includes notes on optimization, algorithms, and more general details that need to be taken into consideration.

This advancement makes us stand out from any existing competition. Our service will take the concept that has caused these apps to take the world by storm, and we will combine it with worker skills and tasks to provide a general user solution to spatial crowdsourcing. Rather than simply having Geographical Information System functionality with a basic simple task, we will have detailed functionality to provide clients a way to get all services they may need in one convenient location. We will also subsequently attract more workers, as we will serve more professions. Workers may also have a better experience on our services than other popular services. This is due to the dynamic spatial crowdsourcing algorithm giving them their optimal job rather than the worker needing to manually find and accept tasks.

4.1.4 Technical Complexity

From a technical side, our project has three main components: the frontend, backend and database. Alongside these it will need to make use of at least one outside API as well, a location API. Our frontend will be composed of two client types, a web browser and mobile, both of these will need to provide the functionality for the user to provide information input to the backend and query the backend data for general profile information, task assignments/updates and location data. Our mobile application will also need to be able to provide location updates for workers to the backend.

1. The backend will need to store all of our data in the selected database. As this data changes it will need to put data through the algorithm and update worker assignments. Part of processing the data through this algorithm will require querying a location API.
2. The frontend will provide functionality to query the backend and provide task assignment updates to workers. On the mobile application side of the front end the application will also need to be able to provide location updates for user reference and also for algorithm usage.
3. Our current target is a more general purpose algorithm. While it is comparable to some existing companies/solutions, like Uber or GrubHub, it is not equivalent, as our algorithm will be taking multiple task types into consideration, as opposed to a single category. This adds an extra layer of complexity to the problem with us having to take more than just spatiotemporal data into account.

Technical aspects that increase the complexity of the project:

- Our project will have three main technical interfaces
    - Backend to database, providing the algorithm with the data that it needs and writing to the database.

    - Frontend to backend, providing user data for the database, location data, and updating users and workers with task info.
    - Backend interfacing with a location api to assist with the spatial dependent portion of assigning tasks.
- The frontend is not one singular component but will need to have components written for both web and mobile.
- ReactJS is new to some members of the team and will require knowledge acquisition.

Internal complexity

Components and subsystems: Technique identification (processing and database),location services, dynamic route optimization

Scientific, mathematical, or engineering principles: Location detection, task optimization and accounting for dynamic situations and changes.

External complexity

Our project has multiple functional requirements that will match or exceed current state-of-the-art standards, such as: error recognition, dynamic location monitoring, task optimization, input processing, and dynamic contingency handling.

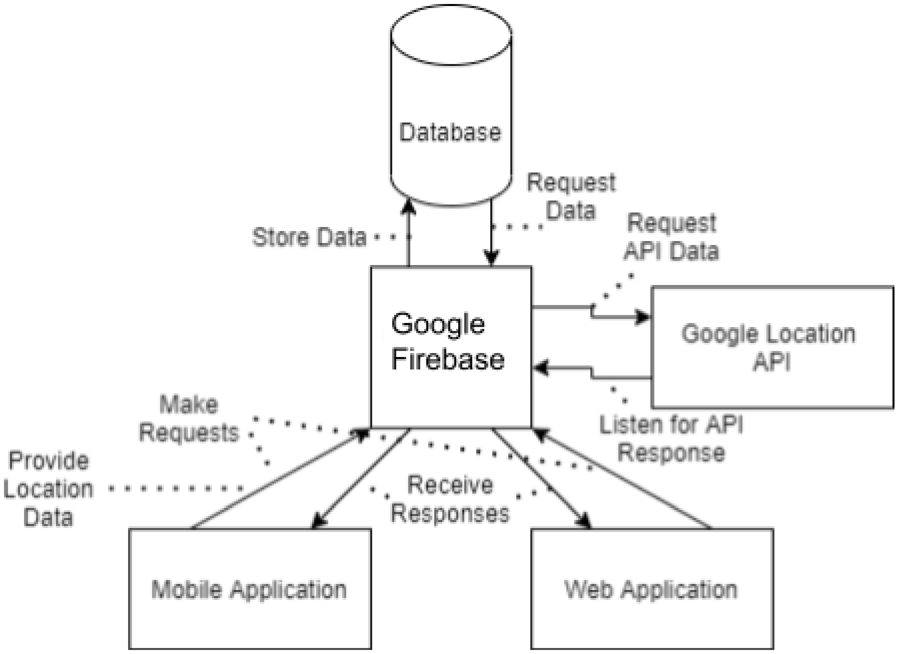


Figure 3: High-Level Architecture Overview

4.2 Design Exploration

Explanation regarding the decision making process behind the application’s proposed design

4.2.1 Design Decisions

Some key design decisions we have made regarding our proposed solution are as follows. Note these may be subject to change with time or at client discretion:

1. Real-time tracking through Google Maps api
2. Front-end implementation using ReactJs
3. We will develop our mobile application using React Native
4. For our backend services, we will utilize the tools provided by Google Firebase
5. Our database provided by Firebase will be used for data storage, using tools such as postman to run tests.
6. The server provided to our development team will be used to host the application's backend.

4.2.2 Ideation

For the backend implementation, we debated between MongoDB (NoSQL) and MySQL (relational). We settled on the use of Google Firebase as the main server side framework. As for DB types, we decided to prioritize NoSQL, and use relational when necessary. These choices were based on finding a balance of ease of use, proficiency, and easily accessible documentation. This platform also has an active community and support teams that can be of assistance if appropriate documentation cannot be found.

For the front-end implementation, we decided to go with ReactJs. We looked into other options such as Angular, JavaScript, Vue.js, and Ember.js. Considering our team's experience, ReactJs was chosen as the best choice for our team. Also, ReactJs is easier to understand and execute than the latter options. One of the cons of choosing Angular is the difficult debugging that it comes with, but we prioritized the performance and efficiency of it.

As for general ideation techniques, these decisions were narrowed down using the information provided in prior sections of this document. We balanced our development team's skills with the needs we had documented, and found the best compromises that suited both categories. Once we settled on a list of possible languages and frameworks, tools such as the weighted decision matrix in the following section were used to make the final decision.

4.2.3 Decision-Making and Trade-Off

Front End Weighted Decision Matrix [Scale: 1-10] :

OPTIONS	Personal Preference	Industry Standard	Developer experience	Ease of Use/learn	Ease of Maintenance	Total
Criterion Weight	20%	10%	25%	25%	20%	100%

ReactJs	9	8	9	10	8	8.95
Angular	6	8	7	9	9	7.8
Vue.js	7	7	8	10	8	8.2
Ember.js	1	3	0	3	5	2.25

Table 7: Design Decision Matrix

As it is shown, ReactJs has the highest total point value on our scale.

4.3 Proposed Design

Detailed descriptions regarding the application's design and necessary functionalities

4.3.1 Design Visualization and Description

Figure 3 *(located in section 4.1.4 of this document)* will be our reference point for the visual description of the design. Our design only includes software components, as such we have a block diagram of the software components and have no further breakdown of any physical elements.

The backend will be our central component taking input from the frontend and pulling data from the database and location api to provide input to our task assignment algorithm. Once tasks are assigned, clients and workers will be able to see tasks and relevant data with elements like worker location updated in real-time and displayed to appropriate parties. The database will be created with Google Firebase, and the front end will be written in ReactJS. Connections between the different elements are labeled in figure 3 with the functionality they will primarily be used for.

4.3.2 Functionality

Functional requirements:

- The mobile and web app should be able to display status of assignment and worker in real-time

  → React allows us to develop both mobile and web apps.

- Only appropriate users should be able to see the location/info of workers

  → Give permission only to appropriate users.

- Interpret existing worker data to decide optimal routes and assignments

  → With the GPS API, collect users' location and use the algorithm to make an optimal decision.

- The applications should be scalable for multiple users

  → Ensure the scalability of the application by choosing the right technology and practices. Using DevOps to manage the project and developing the application.

- Job requests are updated in real time

  → Workers are able to provide regular updates on the request through the UI.

- Workers must be able to post their availability and location

  → Workers are prompted to give their location when posting a job, with real-time data on the jobs they're working.

- Application should calculate the optimal assignment of worker to task

  → Designing an algorithm for assigning the optimal assignment to the worker.

- Users should be able to view status of their requests in real-time

  → Updates made by the worker should be visible at the customer's UI.

- Proper authentication for user accounts

  → Email and Phone Number Verification for all new user Accounts.

- Data storage and tracking of completed tasks

  → All user and assignment data is synchronously stored in the database.

- Optimized for fast response time

  → Ensure industry standard coding practices while using multi-threading for faster response time.

Non-functional requirements:

- Accessible from mobile devices and PC

  → Using React will allow us to develop for both mobile and desktop.

- Design should be extensible.

  → Practice clean code and proper documentation of each task assigned.

- Data should be stored safely and accurately, and its integrity shall be maintained across API hits

  → Store data in the database and use a queue pattern in order not to lose the data.

- Clean and best practice code to improve future maintainability

  → Evaluate team members' performance every other week to ensure all the best practices are being followed.

- Application should be functioning reliably

  → Protects the integrity, availability, and confidentiality of the application and it's users. Also, prepare an emergency plan in case there's a breach.

- Privacy and security of workers

  → Constant testing of our application/code will allow us to find and avoid any data leaks. Use of 2 factor authentication at user login.

- Intuitive and "clean" web and mobile UI

  → Make sure we consult our group with every UI change to prevent clusters in the UI.

4.3.3 Areas of Concern and Development

Concerns:

- Making the application scalable, i.e. adding multiple users and workers over time.
- Making sure that the application is easy to navigate for new users.
- Making the real time location of workers as precise as possible.
- Smooth integration of all the components of the application.
- Making sure the app is optimized for fast response times.
- Making sure the user's information is secure.

Addressing the concerns:

Team will follow a clean coding principle for a smooth integration and merge their code from time to time. As none of the team members has major experience with developing real time applications, we will consult the faculty on ways to mitigate delay within the application. In order to optimize our app for fast response times, we plan to only implement the necessary functions first in order to prevent overcrowding of unnecessary functionalities. Will try to get as much user feedback as possible, on the UI and new user navigation.

4.4 Technology Considerations

The following is our detailed considerations and technology evaluation, and our relevant solutions to any problems discovered in this process.

- **Web Front End:**
  - **ReactJs**
    - Strengths:
      - Re-Usage of Components
      - Easy to Manage due to the modular structure
      - High Performance due to server-side rendering and virtual DOM.
      - Available resources to learn
    - Weaknesses:
      - High pace of development in the framework can be hard to follow up with.
      - Poor documentation due to continuous development
  - Alternatives:
    - AngularJs
    - VueJs

- **Mobile Front End:**
  - **React Native**
    - Strengths:
      - Works on both Android and IOS devices
      - Supports third-party plugins
      - Faster process
      - Pre-build components and reusable codes
      - Easy to learn and set-up
    - Weaknesses:
      - Less scalable
      - Android application can be buggy and might require code changes
  - Alternative:
    - Android Studio

- **Backend Development:**
  - **Firebase**
    - Strengths:
      - Easy to implement
      - Would also host database
      - Has pre-build apis and features to use.

            - Comes with Google cloud
            - Like Realtime Database, Cloud Firestore uses data synchronization to update data on any connected device.
            - Also, offers free credits for new users.
        - Weaknesses:
            - No direct access to the database
            - Offers less support for IOS
            - Less Customizable than other options
    - Alternatives:
        - Spring boot
        - AWS

- **Location API:**
    - **Google Maps Api**
        - Strengths:
            - Potentially useful Android SDK setup regarding Google Maps API
            - Well documented
            - Offers other services such as direction and routing as well.
            - Easy to implement and organize
            - Offers free credits to new users
        - Weaknesses:
            - Not so cost effective
            - Accuracy can be an issue’
    - Alternative:
        - Amazon location services

4.5 Design Analysis

Without the implementation, it is hard to judge the accuracy of the mentioned design plan. However, the practices and technologies mentioned above follow industry standards and align with the use cases of our application. We have carefully considered various design options and made our choices based on various factors such as cost, ease of use, scalability, modifiable, responsiveness, etc. These design choices seem most suitable at this point of time but are subject to change as we progress with the implementation or encounter different challenges.

4.6 Design Plan

Figure 4 represents the application’s structure with respect to the actors and their related use cases. This diagram will be used to ensure the proper requirements engineering process is done, and holds a heavy influence in our technological considerations described in section 4.4

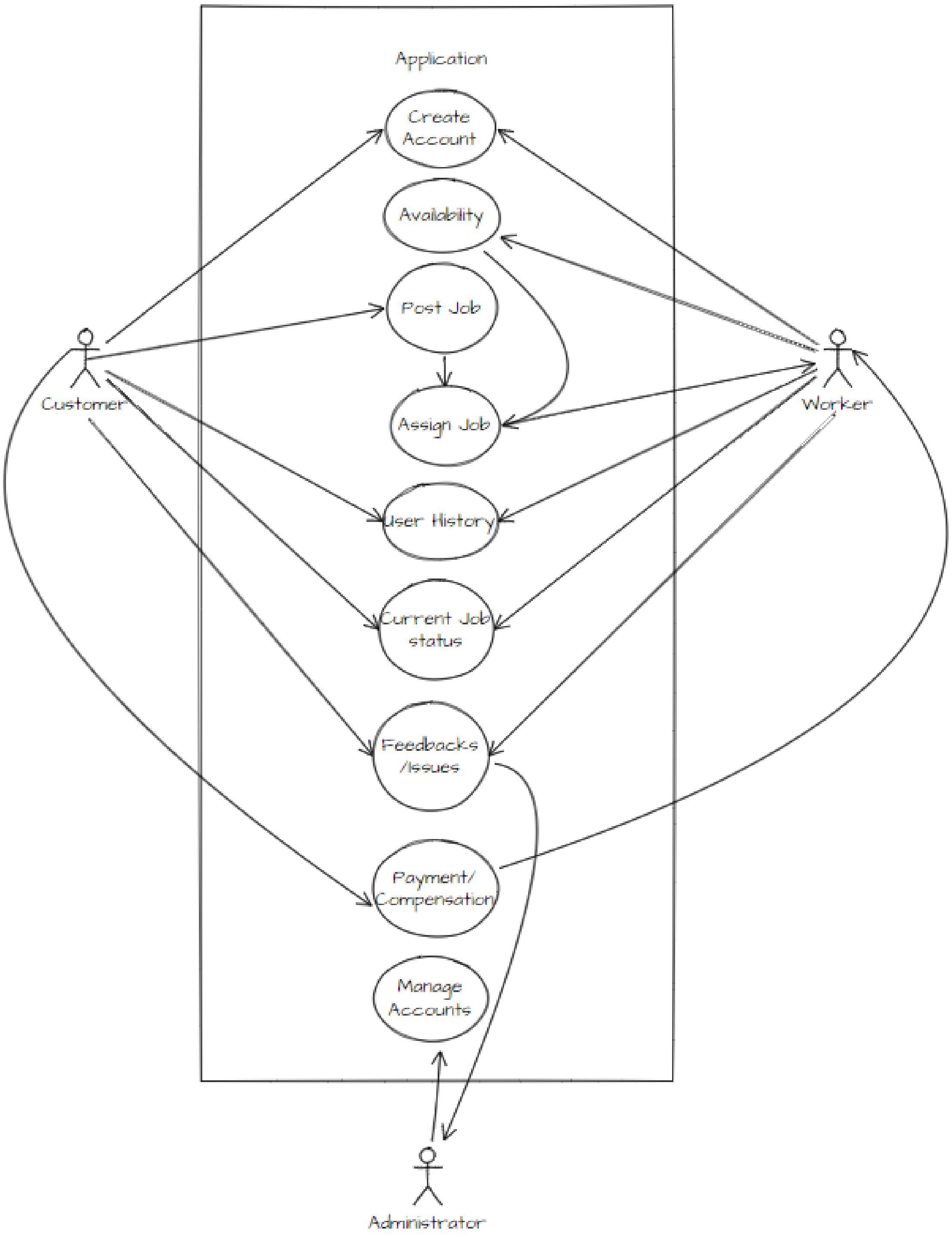


Figure 4: Use Cases

5 Testing

As a part of the Agile development process, testing will be built out and executed as different system components are developed. While minor tests can be done throughout the development process, following the Agile methodology, the main portion of our testing will be done as the final step of that sprint's cycle. This may include, or be followed by a code review by another team member. This system should allow for us to stay organized and systematic, as it is a well-defined and documented process.

A majority of the project-specific challenges we will face throughout the testing process will likely revolve around the Google Location API. Testing this portion of the application will likely require the team to read a large amount of Google's documentation, and to familiarize themselves with the system. We may also then need to mock the API's features in order to test them thoroughly.

5.1 Unit Testing

Our system can be broadly classified into five major modules, i.e Database, Web Application, Mobile Application, Google Firebase Backend, and Google Location API,. For testing purposes, we plan to evaluate each module's functionality and features both separately and collectively. Referring back to Figure 3 *(located in section 4.1.4 of this document)*, within each module there will be different functional units that will collaborate. A few examples of such units are as follow:

- Database module testing.
  - Updates (insertion/deletion)
  - Query procedures.
- API Testing
  - Load Testing using Jmeter
- Google Location API Testing
  - Mock all functionalities needed by the application
  - Use Jest matchers to confirm that the results from the mocks are as expected
  - For these examples, we will have mock transactions to test the correctness of the answer and their overall effectiveness.
- UI testing
  - React user event/Event response
  - Run test in different browsers
  - Accessibility Testing using AMP
  - Run test in different screen sizes

Some of the tools we will utilize for unit testing include:

1. JUnit
2. Mockito
3. TestNG
4. HTML Unit
5. Jest

6. JMeter
7. Mocha
8. AMP
9. Embunit

5.2 Interface Testing

We have two basic modes, mobile and desktop. The basic test for our front end UI will be conducted by running a questionnaire and/or survey with our peers or classmates. The survey will ask participants to identify potential issues with the appearance, functionality, and usability of the UI.

Our primary interfacing will involve interfacing the Google Location API with our Google Firebase backend application. This will involve making calls to the API from our Firebase application and returning spatial information relating to the data that was sent. The data may need to be transformed in a certain way by the application in order to make a proper API call. Our primary method of testing the API calls will be sending specific data of a known, specified location.

Tools:

- Postman
- Mockito

5.3 Integration Testing

Integration testing within the scope of our project will involve CI/CD, or Continuous Integration, Continuous Development. The purpose of our tests, and part of the need for CI/CD, will be to ensure that implementation of new code or software does not negatively impact the working environment. This will involve writing integration tests each time a new piece of software is added, which also ensures that older code will still be tested when new functionalities are added. Most of the critical paths that will need to be maintained involve interaction between each unit. One of the most critical integration paths will involve an end-to-end connection between front and back end; this would entail storing and retrieving task and worker data from the back end per request of the front end. Another critical path will be the connection between the UI map and the location API; translating front end data to data that can be processed with API calls, as well as translating the returned values from the API back into data that can be sent to the front end.

We will check...

Examples:

- How insertion of a new worker and its location is reflected on the map.
- How a report is sent to the server and the corresponding algorithms handle said report.
- Retrieval and storage of incomplete and complete tasks.

Tools:

- Gitlab
- Postman

5.4 System Testing

Refer to the functional requirements found in section 1.2 of the *"Requirements, Constraints, and Engineering Standards"* portion of the project documentation. The following system-level tests may be performed to ensure that said requirements were met:

- When viewing a map or job status page, current and up to date information should be displayed
- When an unauthorized user tries to access information, they should be rejected and redirected
- When a client posts a task, a visual representation is displayed and their task is stored on the server
- Mocked unit tests on location API to ensure data sent and received is as intended.
- Location tests to ensure the UI properly displays a worker's location when needed.
- General flow interface tests will be done to ensure that the movement between portions of the app works as expected
- Tests listed in 4.1,4.2, 4.3 will also fall under the scope of system testing.

The tools used throughout this process may include (but are not limited to): Postman, mockito, Jest, and Junit. We will also use manual and peer reviews to test basic operations and user experience.

5.5 Regression Testing

Sometimes newly added functionalities or changes in existing code make the existing features non functional. And to avoid that, a test suite is chosen that covers both the existing features and newly added ones. And the chosen test suite consists of tests that cover different crucial parts of an application, eg. impact, criticality and most used functionalities. In our case, two such crucial aspects to be tested are scalability and extensibility of the application.

1. We plan on testing the scalability by increasing the traffic on our website and increasing the load on our database testing its impact on the application's overall performance. Apart from load handling, the performance on the user's end will be tested to ensure that the users are getting real-time updates with minimum delays regardless of the fact that the web application or the mobile application is being used.
2. Our program should be extensible and addition of new features in the future should call for bare minimum changes in the existing application. For eg., updating the map information tests its impact on the map services of the app. It has to be ensured that the UI functions properly regardless of the location of the user.

5.6 Acceptance Testing

In order to ensure our functional requirements are met, we will use the demonstration with demo cases. Demonstration with demo cases will help us target specific requirements to ensure the detection of any faults.

In order to test out non-functional requirements, we will merely measure the requirements that are measurable and calculate the percentage of the requirements met to the total number of non-functional requirements.

We will involve the client in the acceptance testing. However, we will also have certain criteria for acceptance testing which the system as a whole must satisfy. A few examples include:

1. How visually appealing the overall UI is.
2. How intuitive the UI is.
3. How efficient and effective the reassignment algorithms are.
4. How extensive the design is.
5. How reliable the application is functioning.

5.7 Security Testing

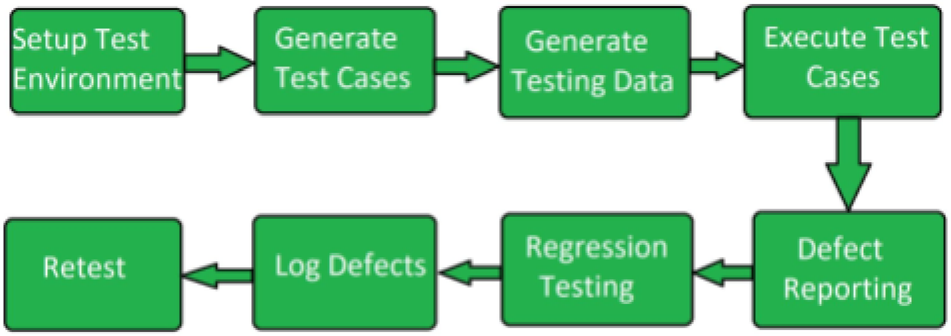
While security is not part of any of the tasks for this project, we recognize its relevancy and we will adhere to it at a system level in terms of password protection, access writes, input sanitization, etc. However, we will not be developing any detailed plans for security testing.

5.8 Results

Throughout the semester we have expanded on different ideas for testing in differing contexts. However, we have done so only in terms of qualitative improvements, not quantitative. There will be other technical improvements as we move through the semester next spring.

One major example of testing proving that the design is as intended is that by using thorough testing we will be able to ensure that our application’s algorithm contains all of the categories discussed in section three of the project documentation. This both helps achieve intended functionality and meets the client’s specifications.

As for a general diagram of our testing structure, for sprints we will be following the Agile methodology. This methodology usually lists testing and reviewing as the 5th and final step of the cycle. For the categories listed in this document, we will be following a process similar to that shown in the figure below.



System Testing Process *(GeeksForGeeks.com)*

Figure 5: System testing process

6 Implementation

Our project is currently going according to the plan laid out in section 3.3. As of now, we have successfully solidified design ideas for both UI and structural implementation, as well as selected application platforms, frameworks, languages, and tools. Because our implementation is so heavily dependent on our design (i.e., we cannot begin development if we do not have the proper frameworks, software platforms, and environments set up). The majority of our implementation will be conducted during the second portion of this class.

As further detailed in section 3.3, our implementation will begin by developing a worker location data controller, optimized such that the data is reflected in real time. We will then begin implementing data controllers optimized such that all important changes should be disclosed to the relevant users in the same real time window. An example of these changes would be the addition/removal of a task or driver.

7 Professionalism

This discussion is with respect to the paper titled “Contextualizing Professionalism in Capstone Projects Using the IDEALS Professional Responsibility Assessment”, International Journal of Engineering Education Vol. 28, No. 2, pp. 416–424, 2012

7.1 Areas of Responsibility

Our application will be following the SE code of ethics. The following table shows an area of professional responsibility, and an example of our applications’ corresponding practices. These examples are followed by a short general summary regarding our beliefs on the subject and provided points.

Area of Responsibility	Definition	NSPE Canon	SE Code of Ethics
Work Competence	Perform work of high quality, integrity, timeliness,and professional competence.	Perform services only in areas of their competence; Avoid deceptive acts.	7.08 In situations outside of their own areas of competence, call upon the opinions of other profession- als who have competence in those areas. 2.01. Provide service in their areas of competence, being honest and forthright about any limitations of their experience and education. **Our summary:** In general, we will only provide services we can safely and accurately do. If we do not know how to do so, we will contact someone who does.
Financial Responsibility	Deliver products and services of realizable value and at reasonable costs.	Act for each employer or client trustees.	3.01. Strive for high quality, acceptable cost, and a reasonable schedule,ensuring significant tradeoffs are clear to and accepted by the employer and the client, and are available for consideration by the user and the public. **Our summary**: We will provide our service at the best quality we are capable of. While we obviously will cut a profit, we will keep prices reasonable to ensure a fair experience.

Communication Honesty	Report work truthfully, without deception, and understandable to stakeholders.	Reports work truthfully, without deception, and are understandable to stakeholders.	3.09 Ensure realistic quantitative esti- mates of cost, scheduling, personnel, quality, and outcomes on any project on which they work or propose to work and provide an uncertainty assessment of these estimates. 5.03 Ensure that software engineers know the employer’s policies and procedures for protecting pass- words, files, and information that is confidential to the employer or confidential to others. 5.09 Ensure that there is a fair agreement concerning ownership of any software, processes, research, writing, or other intellectual property to which a software engineer has contributed. **Our summary:** It is important that we have honest communication with our stakeholders about everything they need to know in order to ensure our project is a success.
Health, Safety, Well-Being	Minimize risks to safety, health, and well-being of stakeholders.	Hold paramount the safety, health, and welfare of the public.	1.07. Consider issues of physical disabilities, allocation of resources, economic disadvantage, and other factors that can diminish access to the benefits of software. **Our summary:** We will follow accessibility standards set in the industry, and will strive to make the application usable by as many users as possible in varying life situations.
Property Ownership	Respect property, ideas, and information of clients and others.	Act for each employer or client as faithful agents or trustees.	5.09 Ensure that there is a fair agreement concerning ownership of any software, processes, research, writing, or other intellectual property to which a software engineer has contributed. **Our summary:** We are committed to ensuring the software and intellectual property used

			throughout this project will be credited to its rightful owner(s).
Sustainability	Protect the environment and natural resources locally and globally.		1.03. Approve software only if they have a well-founded belief that it is safe, meets specifications, passes appropriate tests, and does not diminish quality of life, diminish privacy, or harm the environment. The ultimate effect of the work should be to the public good. **Our summary:** We consider sustainability as an important factor. Even though our product is not directly related to any environmental challenges, we plan to ensure we do no harm to our environment in any indirect manner.
Social Responsibility	Produce products and services that benefit society and communities.	Conduct themselves honorably, responsibly, ethically, and lawfully so as to enhance the honor, reputation, and usefulness of the profession.	1.05. Cooperate in efforts to address matters of grave public concern caused by software, it's installa- tion, maintenance, support, or documentation. 1.07. Consider issues of physical dis- abilities, allocation of resources, economic disadvantage, and other factors that can diminish access to the benefits of software. 2.07. Identify, document, and report significant issues of social con- cern, of which they are aware, in software or related documents, to the employer or the client. **Our Summary:** Presenting well socially is very important for our project. Being a community based project, we must establish trust in our society and in order to do so we must be completely honest about any issues or mistakes that may affect our customers. We must also allocate our resources in a way that doesn't isolate any social group in our society.

Table 8: Areas of Responsibility

7.2 Project Specific Professional Responsibility Areas

- Work Competence
    - Work Competence is a very vital responsibility for our project which we encourage each of our team members to adhere to. Our team has been very consistent with all the deadlines and assignments of tasks. Each team member has presented himself professionally and performed his duties to the best of his capabilities. We work in a collaborative environment and assist each other whenever needed. In areas outside of our competence, we reach out to the professionals to be able to provide fine quality of work.
    - Team Performance : 'High' - We believe that we are performing all of the assignments with the best of our capabilities and on time. We are very well organized as a team and work collectively to build solutions.
- Financial Responsibility
    - Our project deals heavily with the financial responsibility to our client, workers and consumers. Several of our design decisions involved weighing costs for various services in an attempt to keep our project budget as low as possible for our client. For example, when choosing which database service we intend to use, our primary focus was using a service that offered high functionality and scalability at a reasonable price in comparison to others.
    - We also have a financial responsibility to the workers and consumers that will be using our product, as the workers want to get paid a fair wage, and the consumers want fair prices when using our product.
    - As of now our team is performing at a 'medium' level of financial responsibility. This is because although we were able to find relatively cheap options for some of the software services we will be using, there will still be a small fee for our client to cover in order for us to use the services we desire.
- Communication Honesty
    - Our team has been in fairly constant contact with our client, who is also our advisor, and updating them throughout the design process. As of now, our client is our only stakeholder, so our team has been operating at a high level of communication honesty thus far in order to solidify design decisions and ask questions when needed.
- Health, Safety, Well-Being
    - Health: Our project will not necessarily directly apply to the health of the users. They may get a health-related service from the application, but that is not the main use.
    - Safety / Well-Being: As we deal with sensitive information, we have a professional responsibility to protect our users' data. Due to the inclusion of location data this is very applicable to our professional context.
    - Our team believes we are performing at a 'high' ranking on this section, as many above document sections detail our requirements and plans for user safety.
- Property Ownership

    - Our project will provide notice that we do not own, nor are we affiliated with, the services that workers offer on our application. This will serve to protect their intellectual properties and adhere to any legal ownership regulations.
    - Our team believes we are performing at a ‘high’ ranking on this section as well. This differentiation between our company and the services provided by workers is baked into the applications formula, and is always present to users. It also follows practices laid out by existing applications that were intended to complete similar tasks with freelancing service providers.
- Sustainability
    - Our project will maximize our client’s search efficiency by distance and budget which will in return decrease the amount of travel required for the client which leads to less environmental harm. Our project will serve as a supplement to the environment as our goal is to increase economic opportunity, provide safety to users from getting malicious clients, and increase efficiency in carrying out tasks.
    - Our team believes that we are performing at a ‘high’ ranking on the sustainability section, due to the fact that our project goals mentioned above provide protection for the environment and protection for natural resources locally and globally.
- Social Responsibility
    - Our project will reduce car emission by algorithmically calculated optimized routes for the workers which can help the environment.
    - Our team believes we are performing at a ‘high’ ranking on social responsibility, as we are taking care of the environment.

7.3 Most Applicable Professional Responsibility Area

Our team believes that Work Competence is the most important and relevant area of responsibility for our project. As mentioned above, our team has been very consistent in completing deadlines and other deliverables in a timely manner; each team member has also shown that they are competent through their respective works and contributions. Our collaboration with our faculty advisor and client has been productive and beneficial in expediting the design process of our project thus far. The work competence of our team will need to continue to excel as our project progresses through the design phase, as well as into the implementation phase where our team will collaborate even further in software development, implementation, and testing.

8 Closing Material

Concluding details regarding the team and project, as well as extra relevant information.

8.1 Discussion

So far out project work has been focused on research and design decisions. From our research so far we have decided on the main tools with which we shall attack our project. The main tools we plan to make use of include ReactJS/React Native for frontend and mobile development, Firebase for handling our backend, and the Google Maps API for dealing with location data. Before we settled on this plan there was a chunk of research that had to occur to make us as sure as possible that we would not be rushing ahead on our project and unknowingly putting time into a mediocre solution. A part of this research ended up including logistics behind payment methods. While not a large part the use of Google maps and Firebase require some sort of payment option in case we incur some sort of charge.

8.2 Conclusion

So far we have decided on the softwares and resources we will be using in our project implementation. We initially decided on tools which we had some experience with, but gradually realised that some of these tools needed to change based on our use cases. We have also thoroughly researched existing solutions to our problem and read surveys about the algorithm that we need to create. The only minor obstacle we faced could be that some of the decided tools are new for some members which can be learnt and practiced along the way. Overall, the goals that we decided on at the beginning of the semester were met and the team is ready to enter the next phase of this project.

8.3 References

“Contextualizing Professionalism in Capstone Projects Using the IDEALS Professional Responsibility Assessment”, *International Journal of Engineering Education* Vol. 28, No. 2, pp. 416–424, 2012

Tong, Y., Zhou, Z., Zeng, Y. *et al.* Spatial crowdsourcing: a survey. *The VLDB Journal* 29, 217–250 (2020). https://doi.org/10.1007/s00778-019-00568-7

Zhao, C *et. all* (2020) gMission Server [Source code]. https://github.com/gmission/gmission.

Zhao, Y. et. all (2019, June 12). *Destination-aware task assignment in spatial crowdsourcing: A worker decomposition approach*. IEEE Xplore. Retrieved September 22, 2021, accessed from https://ieeexplore.ieee.org/document/8735884.

8.4 Appendices

1. A large portion of our research was based off of the client provided report, "*Spatial crowdsourcing: a survey"*. As this is a licensed and paywalled document, we are not permitted to distribute it in this section. However, it can be found via the 8.3 references section.
2. The secondary links in the cited Git repository's README are a useful resource both for comprehending the concept of spatial crowdsourcing, and for seeing examples of implementation.

8.4.1 Team Contract

Team Procedures

1. **Day, time, and location (face-to-face or virtual) for regular team meetings:**

Team - virtual: *Thursdays 5:30.*
Faculty - virtual: *Tuesdays 4-5pm.*
TA - virtual: *Mondays @ 10:30am*

2. **Preferred method of communication updates, reminders, issues, and scheduling (e.g., e-mail, phone, app, face-to-face):**

Remote via discord text/voice chat.

3. **Decision-making policy (e.g., consensus, majority vote):**

Majority Vote

4. **Procedures for record keeping (i.e., who will keep meeting minutes, how will minutes be shared/archived):**

All the major points/decisions will be noted down and shared across the team on google drive.

Participation Expectations

1. **Expected individual attendance, punctuality, and participation at all team meetings:**

Every team member will try their best to attend all the team meetings, TA meetings, faculty meetings etc,. In case someone can not make it to a meeting, other members will try and bring them back to speed.

2. **Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:**

Roles will be decided in team meetings and will be distributed amongst the team for every task, and every member will be expected to complete the assigned tasks. Every team member will be expected to update the team with their progress and ask for help when needed.

3. **Expected level of communication with other team members:**

All team members are generally expected to keep the team up to date as to the status of what they are working on, or any new and important developments. If you cannot make it to a meeting or if something comes up that will interfere with an assigned task, notify the team as soon as possible.

4. **Expected level of commitment to team decisions and tasks:**
All team members are generally expected to follow assignment due dates and our own personal set due dates very consciously. Team decisions will be by majority vote so whatever is decided is expected to be followed by all team members.

Leadership

1. Leadership roles for each team member (e.g., team organization, client interaction, individual component design, testing, etc.):

1) Cole Dulaney	-	Team Organization
2) Isaac Reed	-	Team Organization
3) Yuichi Hamamoto	-	Testing
4) Aman Agarwal	-	Individual Component Design
5) Shagun Bansal	-	Individual Component Design
6) Seth Platt	-	Client Interaction
7) Abdula Eljaam	-	Testing

2. Strategies for supporting and guiding the work of all team members:
Team members will work both individually and with pair programming in order to thoroughly produce, review, and test code. Will have weekly/bi-weekly team stand-up meetings to discuss progress and future goals.

3. Strategies for recognizing the contributions of all team members:
Contributions of each team member will be recognized at weekly/bi-weekly team meetings. Once development starts contributions can also be tracked by the group git.

Collaboration and Inclusion

1. Describe the skills, expertise, and unique perspectives each team member brings to the team.

1) Cole Dulaney: Full stack web application design and backend development on android applications
2) Isaac Reed: Previous experience with both full-stack web development, as well as working on an agile/SCRUM team.
3) Yuichi Hamamoto: Android development, Unity development, backend development
4) Aman Agarwal: Experience with web application, android application development, and various Testing Tools. Also worked in an Agile/Scrum environment.
5) Shagun Bansal: Experience with web development, Agile environment.
6) Seth Platt: Android Development Experience, Agile development.
7) Abdula Eljaam: Experience in web development and front-end Android development.

2. Strategies for encouraging and support contributions and ideas from all team members:

Allow each member to give their perspective at team meetings, as well as discuss what they've accomplished and what they may be struggling with since the previous meeting.

3. Procedures for identifying and resolving collaboration or inclusion issues (e.g., how will a team member inform the team that the team environment is obstructing their opportunity or ability to contribute?)
This is something that would ideally be brought up in a team meeting, which would lead to team discussion about the presented issue, as well as how it could be solved.

Goal-Setting, Planning, and Execution

1. Team goals for this semester:

Oct. 1 - literature review and scope definition
Nov. 1 - preliminary design; platforms/tools selection
Dec. 1 - final design; testing plan; development/implementation plan

2. Strategies for planning and assigning individual and team work:

Most of the work will be assigned on the skill set of each individual but in a scenario where some new skills are needed, the task will be assigned in a way that the load is distributed uniformly across the team.

3. Strategies for keeping on task:

The main strategy for keeping on task will be constant updates on the task progress for each member. With the constant updates, each team member will be held accountable for their designated task and will be helped immediately if a problem arises. The use of SCRUM methodology also works to ensure that everybody is keeping up to date on their work.

Consequences for Not Adhering to Team Contract

1. How will you handle infractions of any of the obligations of this team contract?
Will try to discuss and solve it within the team. Each team member will try their best to help out whenever needed.

2. What will your team do if the infractions continue?
The team will first try to meet with the individual(s) who are causing the issue and sort it out. If the issue persists, the issue will be brought to a TA/Professor.

Signatures

a) I participated in formulating the standards, roles, and procedures as stated in this contract.
b) I understand that I am obligated to abide by these terms and conditions.
c) I understand that if I do not abide by these terms and conditions, I will suffer the consequences as stated in this contract.

1) Cole Dulaney
2) Shagun Bansal
3) Aman Agarwal

4) <u>Isaac Reed</u>
5) <u>Abdula Eljaam</u>
6) <u>Seth Platt</u>
7) <u>Yuichi Hamamoto</u>