Product Requirements Document

Spatial Crowdsourcing for Dynamic Settings

SDMay22-31

Team Composition: Abdula Eljaam, Aman Agarwal, Isaac Reed, Seth Platt, Cole Dulaney, Shagun Bansal, Yuichi Hamamoto

Original Due Date: 10/3/21

Version 0.8

1.1 PROBLEM STATEMENT

The issue of spatial crowdsourcing involves three main components: tasks, workers, and a platform. In dynamic spatial crowdsourcing, 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.

1.2 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.

1.3 ENGINEERING STANDARDS

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.

1.4 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.