

## **\*Software Requirement Specification\***

**Problem Statement:** A few sentences to describe the problem you are trying to solve, i.e., justify why this software is needed.

With the semester underway, Hopkins' students are extremely busy between classes and extracurriculars, leaving little time to take care of chores (e.g., lawn care, house cleaning, grocery shopping). There is also a plethora of students looking for quick cash, willing to help out with these errands. However, there is no interface dedicated to connecting these two parties, until now!

**Potential Clients:** Who are influenced by this problem and would benefit from the proposed solution? (i.e. the potential users)

Potential clients are Johns Hopkins students. Students who rent property, don't have a car on campus, or just don't have time to do certain things they want done are all potential clients. Upperclassmen may benefit more from the app since they live off campus and don't have the benefits that come with living in a dorm.

**Proposed Solution:** Write a few sentences that describe how a software solution will solve the problem described above.

We are looking to create a Hopkins specific errand ticketing app where users can post a need on an app with specific instructions on what errands they need completed. Then other users, who have more time to possibly make some extra money, can claim these tickets and complete them in a timeframe specified by the creating user. This will allow students to get done that they wouldn't otherwise have time to do. It will also allow other students to make some extra money without the hassle of trying to schedule around an outside job.

**Functional Requirements:** List the (functional) requirements that software needs to have in order to solve the problem stated above. List these in role-goal-benefit format. It is useful to try to group the requirements into those that are essential (must have), and those which are non-essential (but nice to have).

**Must have**

1. As a user, I want to be able to register an account and sign in, so that I can use the app.
2. As a user, I want to be able to view a stream of all tickets (errands/chores), so that I can see which tickets have been created by other users.
3. As a user, I want to be able to create a new ticket so that I can put out tasks that other users can see and complete.
4. As a user, I want to be able to assign a task to myself so that I can pickup the ticket and know that I am responsible for the task.
5. As a user, I want to be able to assign a payment value for the task which corresponds to an agreed-upon amount which describes the difficulty and length of the task.
6. As a user, I want to be able to view all tickets that are assigned to me, so that I can easily see which tasks I am responsible for.
7. As a user, I want to be able to view all of my tickets, which are tickets I have created and are open to other users to pickup, so that I can easily see my created tickets and their corresponding status.
8. As a user, I want to be able to mark a task as complete once I have finished the task and it is ready for review by the task owner.
9. As a user, I want to be able to review my tasks and mark complete tasks as "DONE" once I have verified that the ticket/chore is finished which will send the assignee the payment amount.
10. As a user, I would like to have the option for all CRUD operations on a given task. If I am the owner, I may want to delete/edit tasks I have made to modify descriptions, category, and/or payment amount.

#### Nice to have

1. As a user, I want to have different task categories (such as household chore, shopping/grocery, pet care, etc) so that I can assign a category to the task and later filter tasks by category.
2. As a user, I want to have a feature where I can track the status of a payment so that once the task is completed, I can verify if I have received the agreed upon payment for the errand.
3. As a user, I would like to be able to filter open tickets by price so that I can see compare tasks by difficulty and monetary value.
4. As a user, I would like there to be a chat history feature associated with each ticket so that the owner and assignee can easily communicate on questions related to the task.

Continues on the next page

#### Non-functional Requirements:

#### Performance Requirements:

The system's response time under load should be less than 600 ms to ensure users can browse and submit tasks quickly. The system should handle up to 1000 concurrent users without significant performance degradation.

#### Scalability:

The system should be able to scale dynamically with the growth of university users without major architectural modifications.

#### Security:

User registration and login should use encrypted transmission protocols (HTTPS) and employ OAuth2 or JWT for authentication.

All user data, including payment information, must be stored and transmitted securely with encryption.

Two-factor authentication (2FA) should be implemented to enhance account security.

#### Maintainability:

Development should adhere to standardized coding practices, with detailed documentation provided.

#### User Experience:

Page load times should not exceed 3 seconds.

**Software Architecture & Technology Stack:** Will this be a Web/desktop/mobile (all, or some other kind of) application?

Would it conform to specific software architecture? What programming languages, frameworks, databases, ..., will be used to develop and deploy the software?

The application we want to build is a web application. And we will use the client-server architecture.

We will use Javascript, Node.js, Express.js, React.js, PostgreSQL.

To optimize the performance and handle data consistency, we probably consider Redis, MongoDB and Kafka.

**Similar Apps:** List a few similar applications to the one you are developing. Don't be eager to conclude no similar app exists! There is always something similar to what you are building! Finding those will help you to better specify your project. ***You must be prepared to explain how your app is different from the existing ones.***

- JIRA - JIRA is a project tasking interface allowing users to break down their project into more manageable pieces, rather than posting individual errands anyone can complete.
- GreenPal - GreenPal allows users to post lawn care needs, which local services can bid on, however it is specific to landscaping.
- Spruce - Spruce is for specific neighborhoods where they partner with service providers, rather than on demand independent providers.