Final Year Project Proposal

Name: Krishma Khadgi

Student Number: 2329461

Course: Bachelors (Hons) in Computer Science

University email: k.khadgi@wlv.ac.uk

Supervisor: Mohit Acharya

Date: November 13, 2024

Table of Contents

Table of figures

1. Project Details

Project Title:

Fostering community engagement and service through equitable skill exchange.

Academic Question

• How can live-location based time banking ensure equitable skill exchange, optimize skills exchange and foster community engagement?

Aims

• To encourage skill exchange between users and fostering collaborative and supportive community.

• Promoting community service.

• Making services accessible to marginalized groups.

Objective of the Project

• To facilitate the exchange of services using time hours as currency and ensuring that users can withdraw and deposit time hours by offering or receiving services.

• To implement a platform where services are accessible to individuals regardless of their socioeconomic background or financial resources.

Artefact to be developed

HourFlow is a skill exchange website that promotes community service and collaboration among individuals. This platform incorporates live location-based skill sharing to connect users within a certain range. This platform can be further scaled to include features like: hosting workshops and tutorials for skill development as well as integrating payment gateways like eSewa and Khalti to implement optional monetary exchanges.

2. PROJECT PROPOSAL

Introduction

In a fast-pacing world that is being driven by money and dominated by technology, human connection and collaboration has been diminishing rapidly. The trend of individualism and internet addiction has made it difficult to have meaningful exchanges and cooperation among individuals. Therefore, the platform, HourFlow was adopted to respond to this trend.

By leveraging technology, HourFlow allows equitable skills exchange among the users without any monetary transaction, promoting community service and engagement. The platform aims to bridge the gap between individuals, encouraging a collaborative and service-oriented approach to community building. Additionally, the platform also seeks to address the systematic issue faced by marginalized groups, who lack access to services due to financial barriers. Skill exchanges with time instead of money, creates an inclusive environment where everyone regardless of socioeconomic background can participate and benefit from it.

The academic question focuses on how by incorporating technology, the platform optimizes skills exchange, ensures equitable skill exchange and promotes community engagement. The question also examines how the platform can enhance connection, mutual support and cooperation within local communities, enabling individuals to contribute and benefit from skill exchanges in an efficient way.

Initial Research into sources of information

The initial research for HourFlow involved delving into the concept of Time banking, analyzing platforms that utilize this concept as well as exploring techniques that aid the unique features of HourFlow.

Time banking, the core concept of HourFlow, is a system where people exchange services based on time hours rather than monetary payment, irrespective of the service provided (GivingTuesday.org, 2021). This concept enables individuals to access services without any monetary transaction and facilitate a sense of belonging and collaboration. (Ozanne, Learning to exchange time: Benefits and obstacles to time banking, 2010).

Similarly, the techniques that support the unique features of HourFlow were also explored. One of the core features, live-location tracking utilizes a method in the Geolocation API of JavaScript that retrieves the current geographical location of the user with their permission. The latitude and longitude of their position is provided through an object called ‘position’. We utilize these coordinates to offer real-time location-based service matching (docs, 2024).

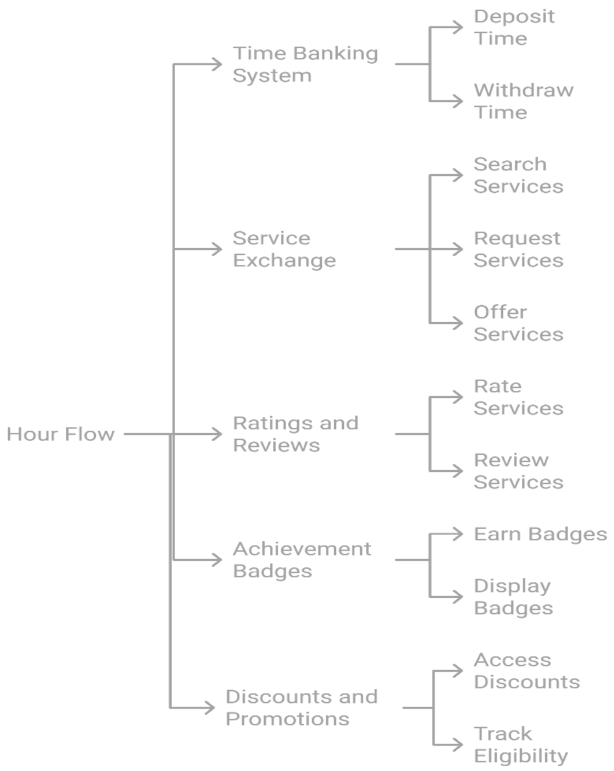
On a similar note, rewards system is one of the cornerstones of HourFlow that enhances user engagement and immersion. The platform provides discounts and achievement badges to keep the user engaged. The data for discounts and badges are stored in the database and updated into the user’s profile when a user meets the criteria for earning a badge or a discount. This gamified approach fosters a sense of achievement, increasing user retention and interaction.

Continuing the initial research, various platforms that utilize the time bank concept were explored and analyzed. Platforms like TimeRepublik, Hourworld, Bank of Time and so on had already employed the time banking system, focusing on creating local time bank system and exchange of services. However, while these platforms did excel in performing core functionalities, HourFlow stood out due to its modern approach and superior technology integration. Unlike these traditional platforms, HourFlow introduced unique features like real-time location skill sharing, discounts and promotions, badges, as well as ratings. These enhancements made the platform more dynamic, engaging and user interactive.

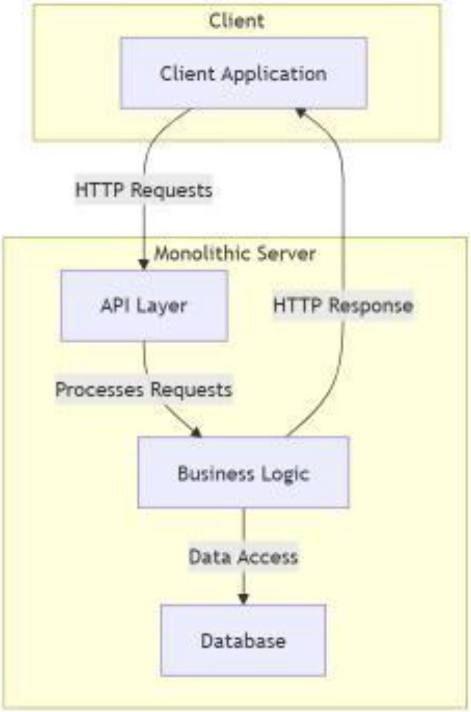
Artefact (proposed)

HourFlow is a website designed to promote community service and collaboration among the users. This is an intuitive platform that utilizes live-location based time banking, ensuring that users collaborate and share skills among a certain radius. The core system of HourFlow, time banking, values all skills equally by measuring contributions in time credits rather than monetary terms regardless of the service provided, ensuring equitable skill exchange. Similarly, the live-location based skill exchange allows users to connect with service providers who are immediately available nearby, ensuring skill exchange optimization and reduce in delays. Likewise, as this platform focuses on geographic proximity, the skill exchanges occur within the community so, this strengthens local connections and promotes community engagement.

HourFlow can be divided into following core components:



Functional Decomposition Diagram 1



System Architecture 1

Benefits for Clients:

- Cost-Effective Services: Clients can access a wide range of services without any financial burden, as they are exchanging it with time rather than money.
- Access to Diverse Skills: The platform connects clients with individuals who possess various skills, allowing them to access a plethora of services, from simple to specialized ones.
- Flexible Scheduling: Clients can arrange services based on their availability and preferences, creating a more personalized experience.
- Quality Assurance: Through the rating and review system, clients can choose service providers based on past performance and quality, ensuring a higher quality of service.

Benefits for Individuals Providing Services:

- Earn Time hours: Service providers can earn time hours that can be redeemed for services they need, allowing them to benefit from the platform without any monetary investment.
- Skill Development: By offering their services, individuals have the opportunity to practice and refine their skills, gaining experience that can be valuable for personal or professional growth.
- Community Engagement: Providing and receiving services without any monetary transaction fosters a sense of belonging and connection to the community, as individuals actively participate in helping others and building relationships.
- Recognition and Rewards: Service providers can earn achievement badges, discounts and ratings, boosting their profiles and showcasing their skills, which can lead to more opportunities and requests for their services.
- Flexibility and Autonomy: Individuals can choose the services they want to offer and set their schedules, giving them the freedom to manage their time according to their availability and preferences.

Future prospects:

- Integration of Learning and Development: Incorporating workshops and training programs where users to earn time credits by participating can promote skill enhancement and lifelong learning, making the platform more valuable.
- Integration of online payment platforms: By incorporating payment platforms like eSewa and Khalti, service providers can also earn monetary rewards for completing a service.

Functional Requirements

1. User Registration, Authentication and Profile Management:

- Users should be able to sign up and create a profile.

- There should be password authentication.

- Users should be able to edit their profile information, including skills offered, availability, and so on.

2. Time Banking System:

- Users should be able to deposit time by offering services and withdraw time by receiving them.

- The system should track each user's time hours and update it in real time.

3. Service Exchange:

- Users should be able to search for available services based on their location and skill type.

- Users can request services from others and offer their services in return.

4. Ratings and Reviews:

- Clients can rate individuals after service exchanges based on their quality, professionalism, and punctuality.

- The system displays average ratings on user profiles to help others make informed decisions.

5. Achievement Badges:

- Users can earn badges for milestones, such as completing a certain number of service hours.

- Badges should be displayed on user profiles.

6. Discounts and Promotions:

- Users with a certain number of hours can access discounts at partner businesses like restaurants, grocery shops and so on.

- The system tracks eligibility for discounts based on time hours.

Non-Functional Requirements

1. Performance:

- The platform should load within 3 seconds for optimal user experience.

2. Scalability:

- The system should be designed to scale easily to accommodate growing numbers of users and services.

3. Security:

- User data must be securely stored and protected against unauthorized access.

- The platform should implement secure authentication methods.

4. Usability:

- The interface should be user-friendly and easy to navigate for users of all technical skill levels.

Plan/Schedule

Sprint 1: UI/UX Design

Duration: 2 weeks

Tasks:

- Wireframe.
- Design user flow and task flow for service requests, time banking transactions, and profile updates.
- Conduct usability testing with initial designs.

Deliverables:

- Wireframes
- Interactive prototypes
- Final UI design

Sprint 2: User Registration and Profile Management

Duration: 2 weeks

Tasks:

- Make login and signup pages, add validation for input fields, password encryption and authentication. Similarly, integrate the database by setting up backend API for user registration.
- Develop profile management features for users to edit skills, availability, and location. Make a frontend profile page where user information is displayed. The information should be fetched and displayed and edited through API endpoints.

Deliverables:

- User registration and profile management module

Sprint 3: Time Banking System (Part 1)

Duration: 1 week

Tasks:

- Develop functionality for users to deposit and withdraw time hours. Build logic for users for depositing and withdrawing time hours. Achieve this by modifying user schema and adding fields to track and store balance and transactions and creating endpoints for updating the user's balance when they offer or receive services.

Deliverables:

- Time banking system module

Sprint 4: Time Banking System (Part 2)

Duration: 2 weeks

Tasks:

- Conduct integration of the time banking system with user profiles. Create a separate dashboard to show transaction and link it in the profile page. The balance will be shown next to personal details. After finishing a service, fetch the data from backend and update it real-time by using hooks.

Deliverables:

- Completed time banking module

Sprint 5: Service Exchange (Part 1)

Goal**:** Implement live location tracking and backend for service exchange functionality.

Duration: 2 weeks

Tasks:

- Use geological API to capture user's live location and set up backend logic to handle service requests and offerings based on skill type and user location. Create API endpoints to allow users to offer and request services and store the location and type of service of the user by updating user schema.
- Implement a method in the backend to filter services by location.

Deliverables:

- Service exchange module

Sprint 6: Service Exchange (Part 2)

Goal**:** Build the frontend for searching and offering/requesting services within a user's location.

Duration: 2 weeks

Tasks:

- Make a search form where users can search through service type. I send a request to the backend with the user's location and search radius.
- There should be form for users to offer or request services within their current location with the user's location automatically attached.
- Able to display a list of services that the backend has filtered based on the user's location and search radius.

Deliverables:

- Fully integrated service exchange system

Sprint 7: Ratings and Reviews

Goal**:** Build the rating and review system for users after service exchanges.

Duration: 2 weeks

Tasks:

- Implement the ratings and reviews system for service exchanges. First of all, update the schema to store ratings and reviews. Then, create a POST API endpoint where users can submit their reviews and build a form to input comments and rating.

- Develop display features for average ratings and reviews on user profiles. To calculate average ratings, create a method in backend to calculate average rating. Then, fetch average ratings and reviews through a GET request.

Deliverables:

- Ratings and reviews module

Sprint 8: Achievement Badges

Goal: Create a system for awarding badges to users based on specific criteria.

Duration: 2 weeks

Tasks:

- Design and implement achievement badges for user milestones. Firstly, set up the criteria for each badge and setup the schema to store the badge information. And to award badges, create a function that checks if a user has met the criteria for earning a badge and update their profile accordingly.

- To integrate badge display on user profiles., fetch the badges from the database and display it in their profile.

Deliverables:

- Achievement badge system

Sprint 9: Discounts and Promotions

Duration: 2 weeks

Tasks:

- Create a new schema in MongoDB to store business details and the discounts they offer.

- Similarly, create a page where users can view all the discounts available from partnered local businesses and use a GET request to fetch the available businesses and their discounts from the backend.

-Implement eligibility tracking for discounts based on time hours. With the field for balance in the schema, use it to compare with the hours required for the discount and work accordingly.

Deliverables:

- Discounts and promotions module

Sprint 10: System Refinement

Duration: 2 weeks

Tasks:

- Improve and refine the system and add details to wherever possible to make them more responsive and user friendly.

Deliverables:

- Completed frontend

Sprint 11: Quality Assurance (QA) & Documentation (Part 1)

Duration: 1 week

Tasks:

- Conduct unit testing for backend functionalities (time banking, service exchange, and user authentication). Use frameworks like Jest or Mocha to write unit tests for backend API routes.

- Conduct unit testing for frontend components. Test frontend components like the time banking dashboard, service request form, and user profile to ensure they display data correctly and interact with the backend properly. Mock backend responses to see if these components behave correctly when receiving data or encountering errors.

-Report writing

Deliverables

- Unit test reports

Sprint 12: Quality Assurance (QA) & Documentation (Part 2)

Duration: 1 week

Tasks:

- Conduct integration testing for service exchange, time banking, and ratings functionalities. Here, we write integration tests that simulate a full-service exchange process from offering a service to updating ratings and time balance.

- Conduct user acceptance testing (UAT) with a small group of users by inviting them to interact with our site and gather feedback on its usability and functionality.

-Report writing

Deliverables:

- Integration test reports

- UAT feedback

Sprint 13: Quality Assurance (QA) & Documentation (Part 3)

Duration: 1 week

Tasks:

- Conduct performance testing for the platform under expected traffic. Here, we stimulate large traffic performing tasks on the site using tools like Apache JMeter or Locust and monitor its response or errors.
- Refine the platform based on feedback and identified issues. Fix any bugs found during the tests and make improvements based on user feedback.

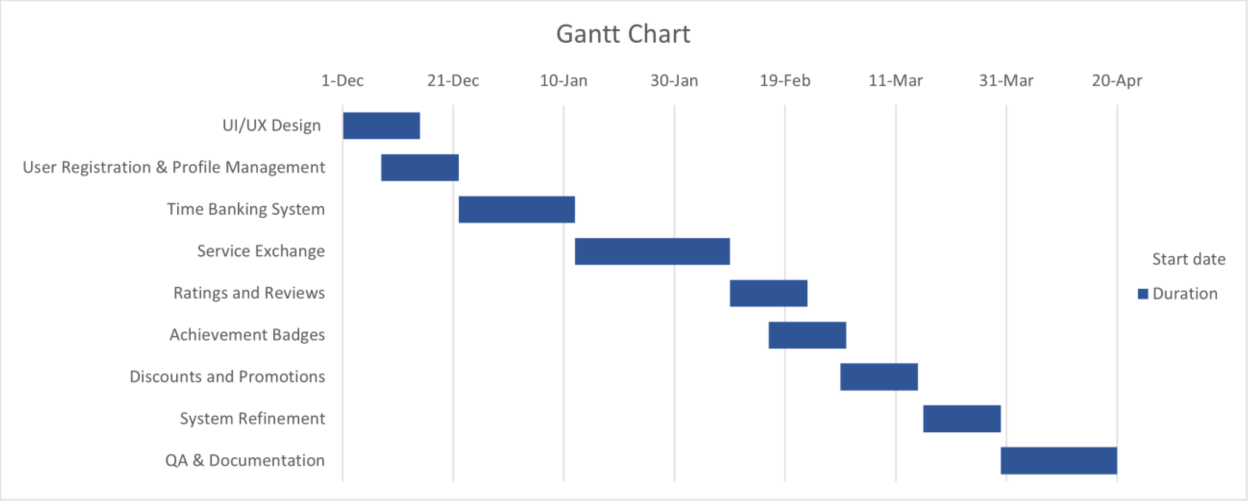
-Report writing

Deliverables:

- Bug and performance reports
- Finalized platform after incorporating feedback
- Report

Timeline table for HourFlow

Sprints	Start date	Duration
UI/UX Design	1-Dec	14
User Registration & Profile Management	8-Dec-24	14
Time Banking System	22-Dec-24	21
Service Exchange	12-Jan-25	28
Ratings and Reviews	9-Feb-25	14
Achievement Badges	16-Feb	14
Discounts and Promotions	1-Mar-25	14
System Refinement	16-Mar-25	14
QA & Documentation	30-Mar	21



Gantt Chart 1

References

docs, m. w. (2024). *Geolocation API*. Retrieved from mdn web docs: https://developer.mozilla.org/en-US/docs/Web/API/Geolocation_API

GivingTuesday.org. (2021). *Timebanking: Building a Stronger Community.* Retrieved from GivingTuesday: https://www.givingtuesday.org/generosity-toolbox/time-banking-building-stronger-communities-one-hour-at-a-time/

Ozanne, L. K. (2010). Learning to exchange time: Benefits and obstacles to time banking. *International Journal of Community Currency Research*, 16.

3. ADDITIONAL INFORMATION

Resources:

Frontend:

- React: A JavaScript library for building user interfaces, allowing for efficient management and reuse of UI components.

- Tailwind CSS: A CSS framework made up of utility classes which makes it easy to build modern and responsive layouts.

Backend:

- Node.js: A runtime environment that executes JavaScript on the server side, and enables the creation of scalable and efficient network applications.

- Express.js: A minimalist framework for Node.js that allows us to create scalable web applications.

- Mongoose: An ODM for MongoDB and Node.js that facilitates data structuring and schema validation.

Database:

- MongoDB: A flexible NoSQL database ideal for storing user profiles, service exchanges, and time banking transactions.

Authentication and security

- JSON Web Tokens: To handle user authentication.

- Bcrypt: To hash passwords for security.

Development Tools

-VSCode: Code editor to write codes for the platform.

- Postman: For API testing.

-GitHub: Version control platform to manage HourFlow's code base.

Client: Mohit Acharya