Final Report

Life Giving Water International

Agua Vision

Team: Brian Huynh, Daniel Chae, Tyler Bylsma

Advisors: Chris Visscher, Mark De Haan

Project Description

An app that uses OCR to read water meter readings and charge households accordingly. The app needs to be able to function offline as connectivity is unreliable and must be able to keep data of water usage for hundreds of families and households across multiple communities.

Vision Statement

LGWI is a non-profit organization that provides water for communities in Ecuador. We're creating an app that helps LGWI measure how much water each household actually uses so people can pay for the amount they consume. Before this, everyone in the community paid a flat rate of about a dollar each month, which isn't fair since some families use very little water while others use much more. Because the current system relies on handwritten meter readings, our goal is to build an app that can read these meter levels accurately using OCR and then charge each household correctly based on their true usage.

Background

Since 2015, LGWI has provided clean water systems to more than 2,000 families. Over the years, they have added pumps and gravity-fed water systems to communities across Ecuador, pursuing their mission to bring clean water to those without access. LGWI continues to maintain and manage these systems, relying on both donations and user payments. To support this, communities track usage with simple analog water meters, and LGWI uses these readings to determine each household's monthly bill. Because the readings are handwritten and sometimes inconsistent, LGWI partnered with Calvin students to develop an application that can reliably collect and store meter data. An app, rather than texting in readings, allows for offline functionality, image capture, OCR accuracy, and secure long-term storage, all of which reduce errors and standardize the process. The app must function offline, accurately record and monitor water usage, and securely back up data to maintain long-term records. Our team is working to design a sustainable solution that meets LGWI's needs now and into the future.

Normative and Ethical Considerations

For each of the design norms, describe a potential consequence of ignoring that norm in your project, and either what specific decisions you might make to align with that norm or challenges you might face in doing so.
Summary of "Design Norms" for Technology

- Cultural Appropriateness
    - We are still struggling with developing an app that is culturally appropriate, and this involves making a Spanish version of the app. This is something we are beginning development on.
- Transparency
    - We have noticed that the OCR can sometimes incorrectly record the correct meter reading, which is something that we have worked on, but it could use a little bit more fixing, so that it works correctly every time.
- Social
    - Someone could try and use the app maliciously, to purposely charge neighbors more than they actually should. So we will have to have trust in the "water person" to do their job honestly.
- Stewardship
    - This application is going to use technology, specifically a phone's camera. We have to ensure that the resources needed for this application are sustainable to different areas that will be using it.
- Aesthetics
    - We have created an aesthetically pleasing application that is intuitive for those with little technology experience. Also, we are looking to do user-testing to make our application easy and intuitive.
- Justice
    - Charging people for what they actually use, make sure that they are not paying more or less, if not, communities may have to go back to charging everyone the same rate.
- Caring
    - Our app provides care for users through its intuitive user interface and adaptive language options.
- Trusting
    - Our app is dealing with pricing of water, and we need to make sure the functionality of any calculations that are happening are 100% accurate. Our app automatically calculates the price that needs to be charged by using the price rate of water and the amount of water used since last reading.

Product Success Criteria

Criteria	Minimum	Stretch
Provides the ability for community system water operators to quickly and efficiently input water use data from local water meters.	Manual input	Manual input and OCR reading
Provides the ability for community system administrators to accurately charge users for water use and provide receipts for the transaction.	Maintains data from previous time period to allow for calculation of water use and ultimately cost for the billing period with the ability to confirm receipt of payment by the administrators	Maintains water use and payment data across an extended period (say 1+ year) which could eventually be printed and kept offline
Stores water use across multiple communities to allow LGWI and CODEINSE the ability to track and compare water use across different timelines and in different regions.	Maintain an easy-to-read database which holds water use data for each home across multiple communities (provided the community consents to this data being held)	Allows the ability to connect GPS location to the homes (producing a file which can be uploaded into a GIS program) including water

Development Process

We are using Github Flow and are meeting weekly to discuss what we have done in the past week, and what we plan on working on for the upcoming week. We are doing a mix of individual and group work. When working individually on certain tasks that relate to the app, such as a certain feature or functionality, we develop, create a branch and push our changes, then someone from our team will review those changes and merge them into main. Show progress to our advisors, and keep them updated.

Steps of our process so far:

- Meet with advisors and get an idea of what they are looking for.
- Figure out our Tech Stack for this project.
- Design the app UI and layout in Figma.
- Development of the app begins, working on pages and overall layout.
- The database is built and connected to the app.
- Full CRUD is implemented into our app.
- Features such as OCR and Local-First Storage are in development.

Challenges we have encountered:

- OCR readings being inconsistent.
- Local-First database implementation.
- Needing offline OCR because of limited internet access.

Team Contract

Expectations

The success of the group will depend on the cooperation and professionalism of its members. Effective collaboration includes, but is not limited to:

- Participating fully and in good faith in both group discussions and project work.
- Participating professionally (civil discourse, academic honesty, and respect for all voices).
- Meeting responsibilities by completing assigned tasks on time and to the best of your ability.
- Being punctual to meetings and notifying the team in advance if running late.
- Acknowledging that workloads may increase near project deadlines, and committing to contribute more time during those peak periods.
- Notes will be stored in a shared Google Drive folder for accountability.
- Communicating conflicts (e.g., interviews, sickness, competitions) early so the team can redistribute tasks fairly.

If unexpected commitments arise, team members will:

1. Notify the group as soon as possible.
2. Adjust responsibilities temporarily
3. Ensure critical deadlines are still met.

Division of work: Tasks will be divided based on skills, interests, and workload capacity, but everyone will contribute to both technical and non-technical parts of the project.

Communication

- Platforms: The team will use Discord? for real-time communication and email for formal updates.
- Response time: Members are expected to respond within 24 hours on weekdays and to confirm receipt of important deadlines.
- Notifications: Everyone will test and confirm notifications are enabled on chosen platforms.
- Accountability: One designated member will check in with teammates who are unresponsive (“ghosting”) and ensure regular communication.

Meeting Schedule

- Frequency: The team will meet once per week for 1 hour at a recurring time agreed upon by all members.
- Modality: Meetings will be held in-person/Teams.
- Flexibility: Extra meetings will be scheduled as needed around key deadlines.
- Attendance: Members are expected to attend all meetings unless they notify the group at least 24 hours in advance (except in emergencies).
- Meeting management: A rotating system will be used where each week one member leads the meeting and another takes notes

Equitable Contribution & Conflict Resolution

- Contribution: Each member is expected to contribute equally.
- If a teammate contributes too little:
    1. The issue will first be addressed privately with the teammate to clarify expectations.
    2. If the issue persists, it will be brought up in a group meeting for open discussion.
    3. If the teammate still fails to improve, the group will escalate the issue to the instructor.
- Conflict resolution steps:
    1. Address the issue directly and respectfully within the group.
    2. Propose actionable solutions (redistribute workload, clarify deadlines, etc.).
    3. If no resolution is reached, the team will document the issue and involve the instructor for mediation.

Deadlines & Accountability

- Internal deadlines will be set before official deadlines to allow for review and revisions.
- Members are expected to complete work to a reasonable standard of quality, not just completion.
- Each member is responsible for communicating if they cannot meet a deadline as early as possible.

Workload Flexibility

- The team recognizes that workload may fluctuate due to external commitments.
- If a member is overloaded, tasks may be redistributed to ensure fairness and timely progress.
- Workload distribution will be revisited regularly to maintain balance across the semester.

Development Environment

To effectively build and maintain the water meter reading application for Life Giving Water International (LGWI), we have chosen a development environment that supports mobile development, offline capabilities, and team collaboration.

Tools and Technologies

1. IDE and Code Management

- **Visual Studio Code (VSCode)**: Our primary integrated development environment (IDE) for writing and managing code across platforms.

- **Git & GitHub**: For version control and collaboration. We use the Git Flow workflow to manage branches (e.g., feature, development, and main branches), allowing each team member to work independently and merge changes systematically.

2. Mobile App Development Framework

- **React Native (JavaScript/TypeScript)**: Chosen for its ability to build cross-platform mobile applications from a single codebase. This enables deployment on both Android and iOS devices, which increases the accessibility of the app in rural Ecuadorian communities.
- **Expo**: Used to simplify React Native development, testing, and deployment—especially helpful for rapid prototyping and device testing.

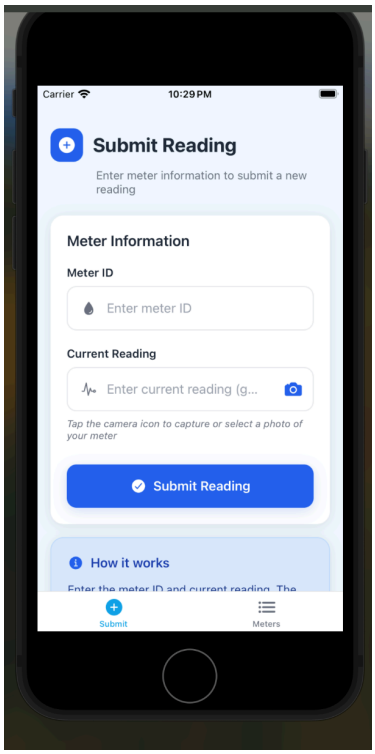
3. Data Storage and Synchronization

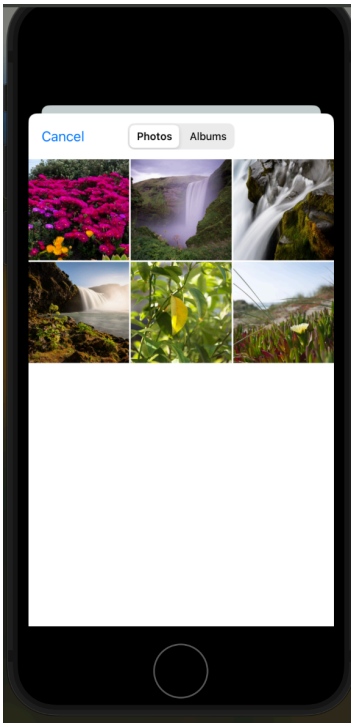
- **SQLite**: For local data storage on the mobile device, ensuring offline functionality and data persistence between readings.
- **Supabase (Optional / Stretch Goal)**: If connectivity becomes more consistent in the future, Firebase can be used for secure cloud backup and real-time syncing between multiple devices.

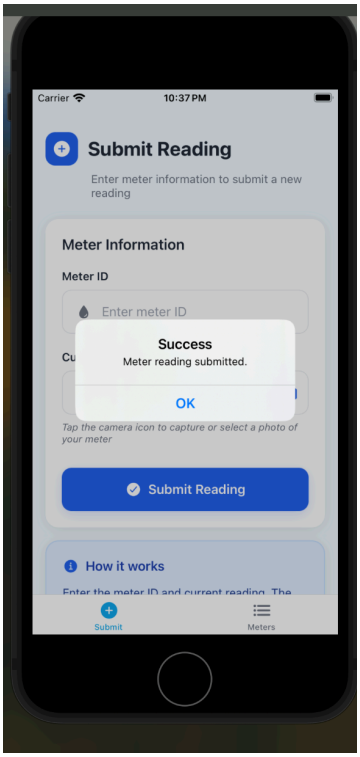
Accomplishments (so far)

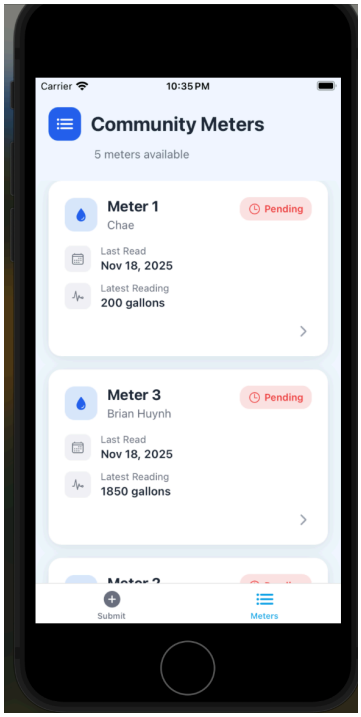
- Full CRUD operations connected to SupaBase.
- Initial app design and implementation completed.
- OCR feature for meter submission.
- Local-first database implementation began.

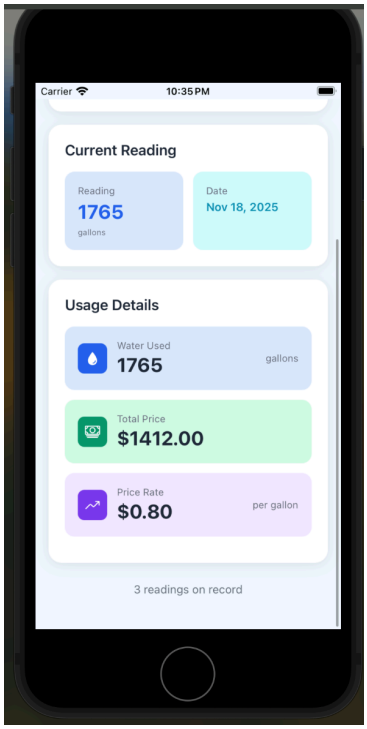
Results (screen shot of working app so far)











Milestone-Specific Section

We updated our Vision Statement, Background, and Normative and Ethical Considerations sections based on the feedback given in Milestone 2 and requirements for Milestone 3. We also added Accomplishments and Results sections, as well as updating our development process section.

We would say that we are progressing well compared to our success criteria. Overall our app is functionally working well, and we have completed the overall app design. We were able to implement both the manual input and OCR feature for inputting the meter readings. While there still needs to be some improvements to our OCR feature, overall we are progressing well in this criteria. We have implemented a system where the app calculates the charge for each meter, based on the price rate of water, and the amount of water used since the last meter reading. So I think we are doing a good job progressing in this success criteria as well. We have a well organized database setup through Supabase and are currently working on a local-first database within the app for offline use. However, we do have to actually get the specific communities into the database and setup, and we plan to do this by creating an administrative version of the app, where the administrator can add and remove communities accordingly. So, overall I would say we are progressing well and have successfully implemented most of the minimum requirements, and will look to complete these minimum requirements and move on to the “stretch” success criteria.