5G Cellular Communications Network Performance

Team: 5G-PAWS (Nick Grosskopf, Sasha Persaud, Sakai Alexander, Jake Waclawski, Patrick Sorensen)
Sponsor: Clark Hochgraf - Communications Research Group, RIT ECET department
Coach: Kenn Martinez

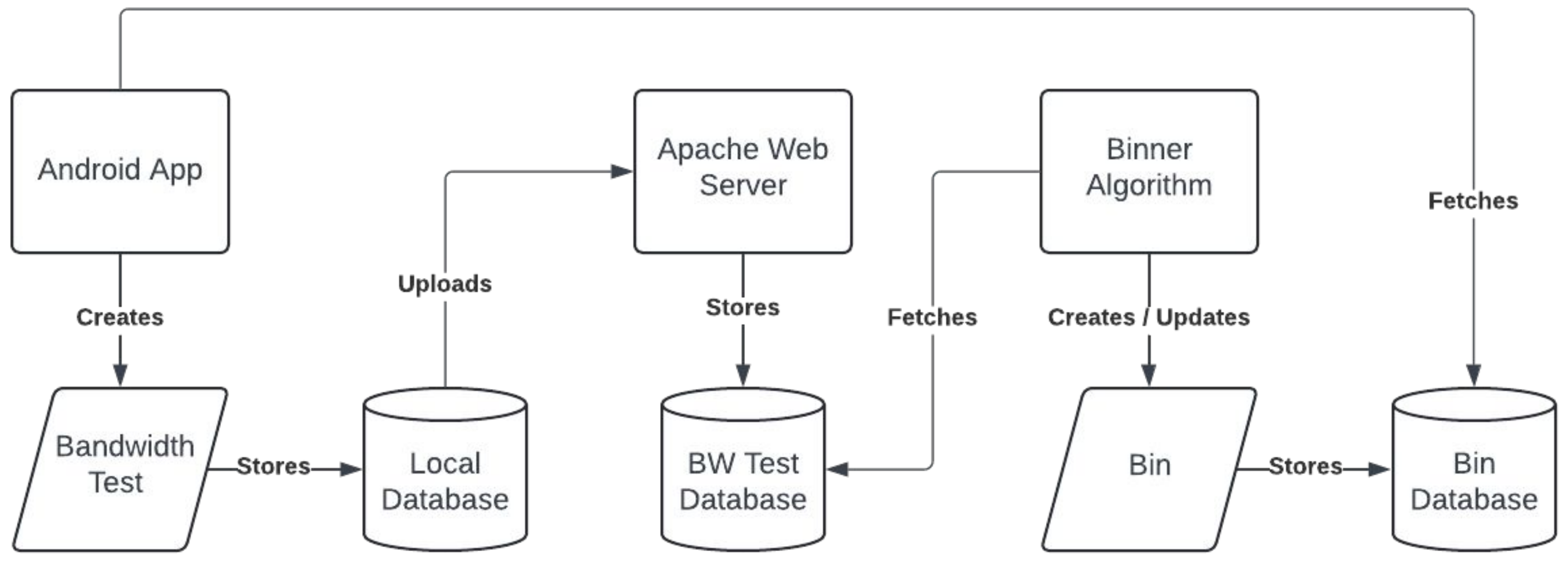
Software Engineering
Rochester Institute of Technology

Why Cellular Metrics?

Cellular Metrics provides access to crowdsourced cellular network coverage maps. This enhances first responder communications by increasing visibility of LTE and 5G networks, especially in disaster scenarios where cell towers may fail. Cellular Metrics uses a system that collects real-time data on network performance via a smartphone app, servers, and a database. The app presents this data on a map to show users coverage strength in different areas. The ultimate goal is to create a scalable, efficient system for monitoring and optimizing cellular communication, especially during emergencies.

System Architecture

The architecture of the Cellular Metrics system was defined by previous senior project teams. Our team's task was to improve upon this existing system and add new features to it. The entire system is composed of an Android App which communicates with a web server with accompanying databases.



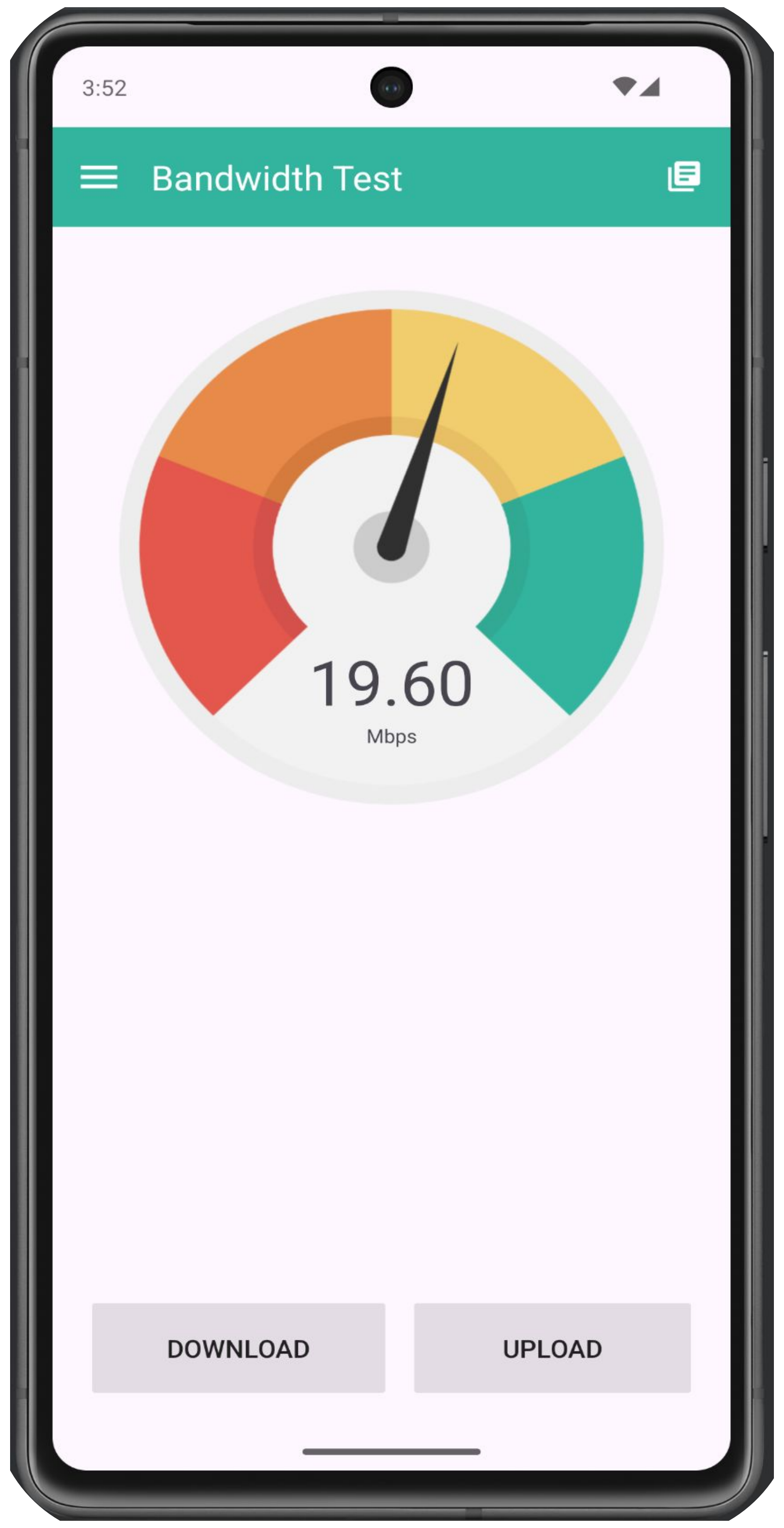
Technology Stack

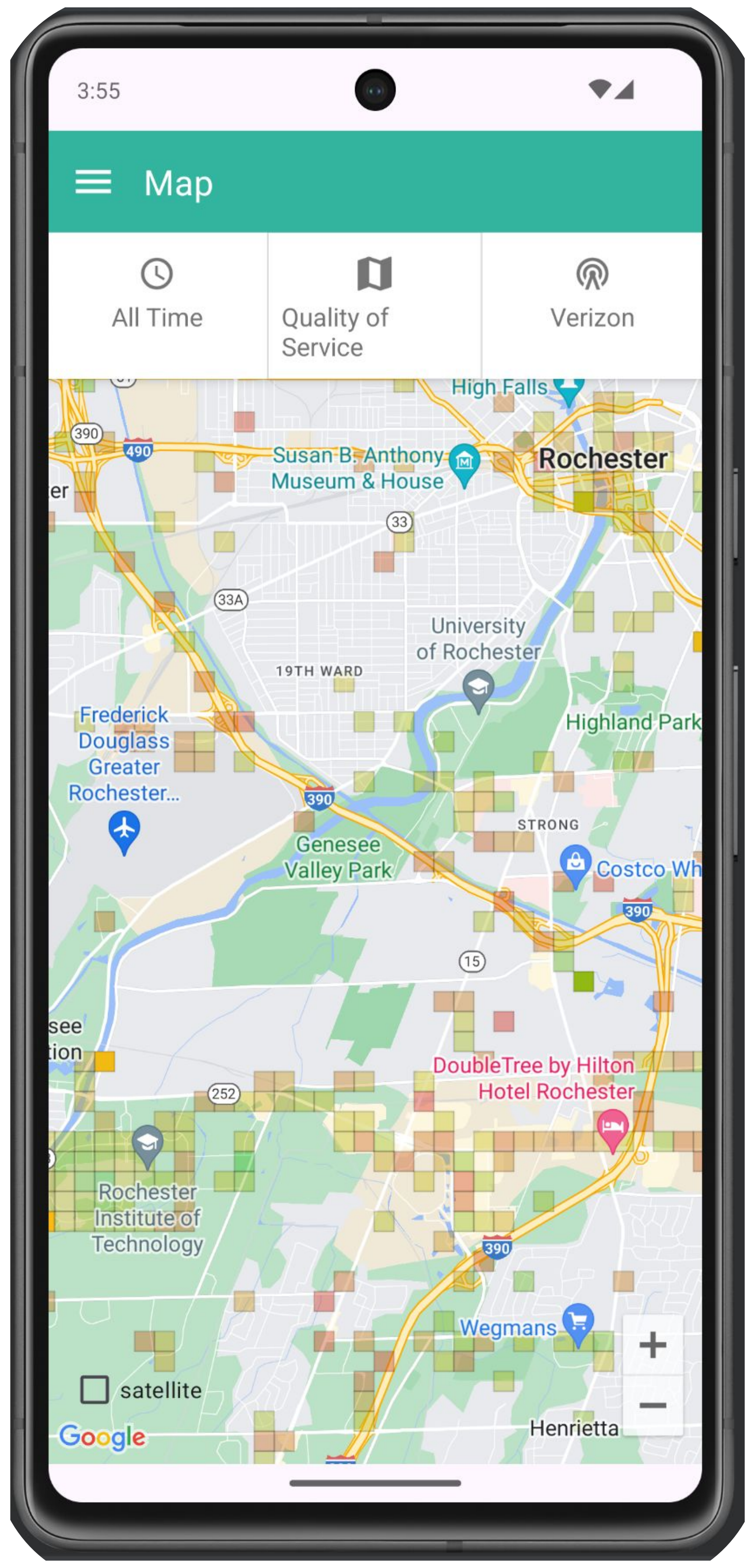
Java
Android App implementation

Android Studio
Android App development tool

Python
Web server implementation

PostgreSQL
Backend database language





Major Outcomes

We've greatly improved the project's maintainability with the creation of thorough documentation of the system, including an architecture diagram, a binner algorithm diagram, roadmaps for learning each of the project's tools, and onboarding instructional documents for every major component of the system. The system has been updated and readability has been improved. With the tools we've left, future teams can now easily learn the system and begin development.

Users can now understand why Cellular Metrics needs certain permissions, and can easily grant access to them with our new popup windows. Pre-processing of new bandwidth test data now runs over 300% faster given our improvements to the Binner algorithm. Furthermore, phones using a 5G network can now send 5G-specific data to the server, giving users even more insight to the coverage in their area.

Process & Methodology

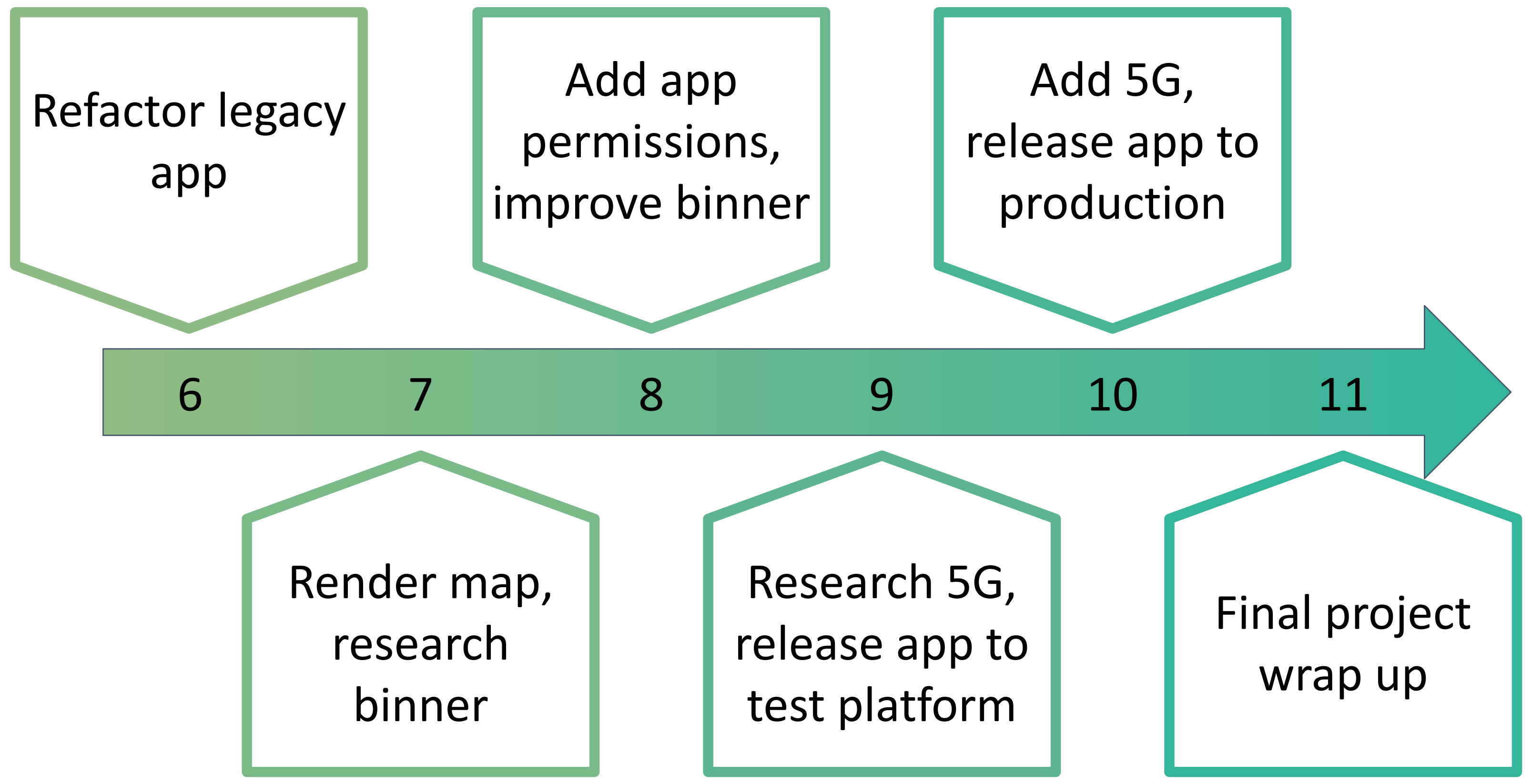
Our team decided on using the agile Scrum methodology for its incremental process and quick adaptability. We performed two-week-long sprints, each beginning and ending with sprint planning and retrospective meetings. We used the Jira Software to aid in our Scrum processes, which allowed us to create and track tasks within each sprint. To help specialize and be more efficient as a team, we split up into two sub-teams, a server-side (Sasha, Jake) and an app-side (Nick, Sakai, Patrick).

Fall Semester | Sprints 0-5

The project was not ready for development when we received it. We spent these sprints uncovering the effects of a few years without maintenance, researching modern best practices, and developing plans to refactor the system.

Spring Semester | Sprints 6-11

Entering the second semester the team had a much better structure. A sprint roadmap was created and followed through, focusing on feature implementation, documentation, and releasing the application to the Google Play Store.


Refactor legacy app
Add app permissions, improve binner
Add 5G, release app to production
6
7
8
9
10
11
Render map, research binner
Research 5G, release app to test platform
Final project wrap up

Future Work

- iOS App port
- Support for network type map filtering
- Crash analytics and unit testing
- Real time map updates with new bandwidth test data
- Support landscape mode for use on tablets