**Implementation and Justification**

**Phase 1 Implementation**

Phase 1 focuses on building a Tracker App to provide a secondary method on data collection. We chose to build an App because only App can run in mobile’s background to collect data passively. Flutter is chosen because our team find it easy to pick up compared to other languages. Having a custom build app can also better to carter client’s need, allowing us to implement a version of activity diary and location survey that focuses on getting crucial information only. Due to time constraints, our team aims to implement these features without any server unless the project gets completed much earlier than expected.

**Feature 1 - On first app launch, the Tracker App user will be prompt to create User Profile. (Require user’s Garmin account). User Profile’s data will be uploaded to cloud storage on changes.**

Reasons: User information is required to identify respondent and to understand respondent profile. Some crucial information is vehicle ownership, age, gender etc. Respondent run app in own device only. Does not require authentication.

**Feature 2 - With user’s permission, app should be able to collect location information. (1 minute interval) -with proper ethics protocol. Location data will be stored locally in the user’s handphone.**

Reasons: Relying location data solely from smartwatch might have too much information loss. Data file’s size is small. This can be deployed without building a server.

**Feature 3 - On app launch, users will be guided to add frequent places for a list of activities.**

Reasons: “Mode of transport” for these frequent places can be requested here. This provides crucial information for frequent locations and reduces ambiguity for certain significant places with multiple activities.

**Feature 4 - Reuse elements from feature 3.**  
  
**Reasons:** Reduce code redundancy. Features 3 & 4 are mostly similar.

**Feature 5 - Data will be sent to cloud storage periodically and data will be deleted periodically.**

**Reasons: This can be deployed without building a server. Researchers can easily get access to these data by logging into the** cloud storage account.

**Phase 2 Implementation**

Phase 2 focuses on building PC software to carry out tasks such as data integration, adding information into existing data. While our team design this project, we notice that most of this task can be better carried out by PC than mobile. Instead of processing the data in respondents' mobile phone before sending data to researcher, we aim to send data to the researcher with minimum processing and offload these to researcher’s PC. Doing so would reduce the need to reliance on respondent’s mobile processing power. Besides, batch processing can be easily implemented with PC software and hence it won’t add too much work to the researcher. We aim to build PC software to help researchers combine multiple tables into one table with as little steps as possible. For Phase 2 Data Integrator software, we choose to use Python because it has a robust library for file processing and GUI building. Python also happens to be one of our team’s most familiar languages. Our team also learned ways to integrate SQL database in Python application. With SQL database, we can easily combine tables and do row operations.

**Feature 6 - Label modes of transport by referring to information from “Frequent Places” and speed.**   
  
**Reasons:** If a respondent is visiting their frequent places, we can use mode of transport that user first key in as a reference. For other locations we can do some guess work for the mode of transport by observing the speed. Although relying on speed to estimate the mode of transport is less accurate, this would cause fewer problems if most of the respondent trips were in between their frequent places.

**Feature 7 - Labels of significant places will be provided as a separate table and stores in researcher’s database.**

Reasons: With longitude and latitude of user’s location, these labels can be easily added by merging these two tables in researchers PC. (e.g., if longitude is between -31.95 to -31.96 and latitude is between 115.86 to 115.87, add label “Shopping Mall”.)

**Feature 8 - Identify and log any anomalies in the data or missing values in Data Integrator software.**

**Reasons: This can be** done easily with PC software by setting simple rules and logic to highlight rows that lack crucial information.

**Feature 9 - Integrate the above streams into a dataset output with SQL in Data Integrator’s backend.**

**Reasons:** Our team members are all familiar with SQL language and SQL is very intuitive in processing tabular data.