1. Proposed software architecture

1.1. Overview

1.2. Subsystem decomposition – Identify the subsystems and the responsibilities of each. You should use component diagrams.

See UML diagram under 1.6.

1.3. Hardware/software mapping – How will subsystems be assigned to hardware? You should use deployment diagrams.

Once the Zebra API registers onClick event, everything else will be taken care of by the provided barcode.

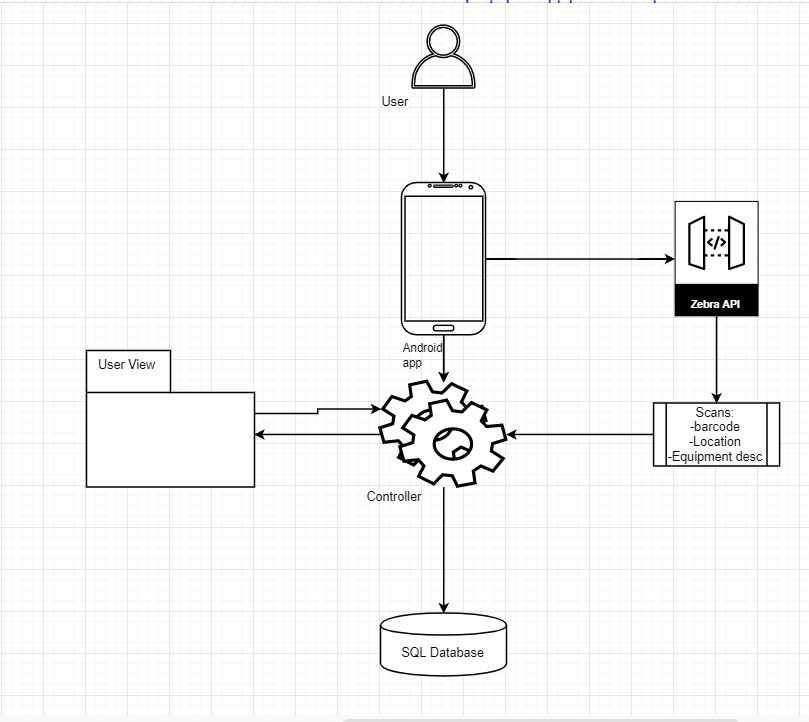
1.4. Persistent data management – Identify the data which will be persistent. Describe the file system or database to be used, including a complete database design.

We will implement a table with barcode as the primary key. In addition, we will keep track of the product name, location, quantity, and last update date

1.5. Access control and security – For each different actor (user, system administrator, etc.) describe the operations they will be enabled to use. Describe authentication and security provisions.

At the moment, there will only be administrative access to the application requiring a login and password.

1.6. Global software control – Describe the control flow (e.g. procedural, event-driven, threaded). Procedural control flows should be described using activity diagrams. Event-driven flows are best described using sequence and state diagrams (use UML diagram standards).



1.7. Boundary conditions – describe how the system will be started up, initialized and shut down. How will it respond to errors and exceptions? Any daily/weekly/monthly/yearly efforts necessary? All organizations buy new computers every 3-4 years, how to migrate to new server? Be able to bulk dump all data to file and bulk load all data from the same file.

The app will be packaged as a .apk file and installed on the Zebra device. The database connection will be configured in the source code. Database migration will require reconfiguration of the connection on the device. There will be error/exception handling for user/connection errors.