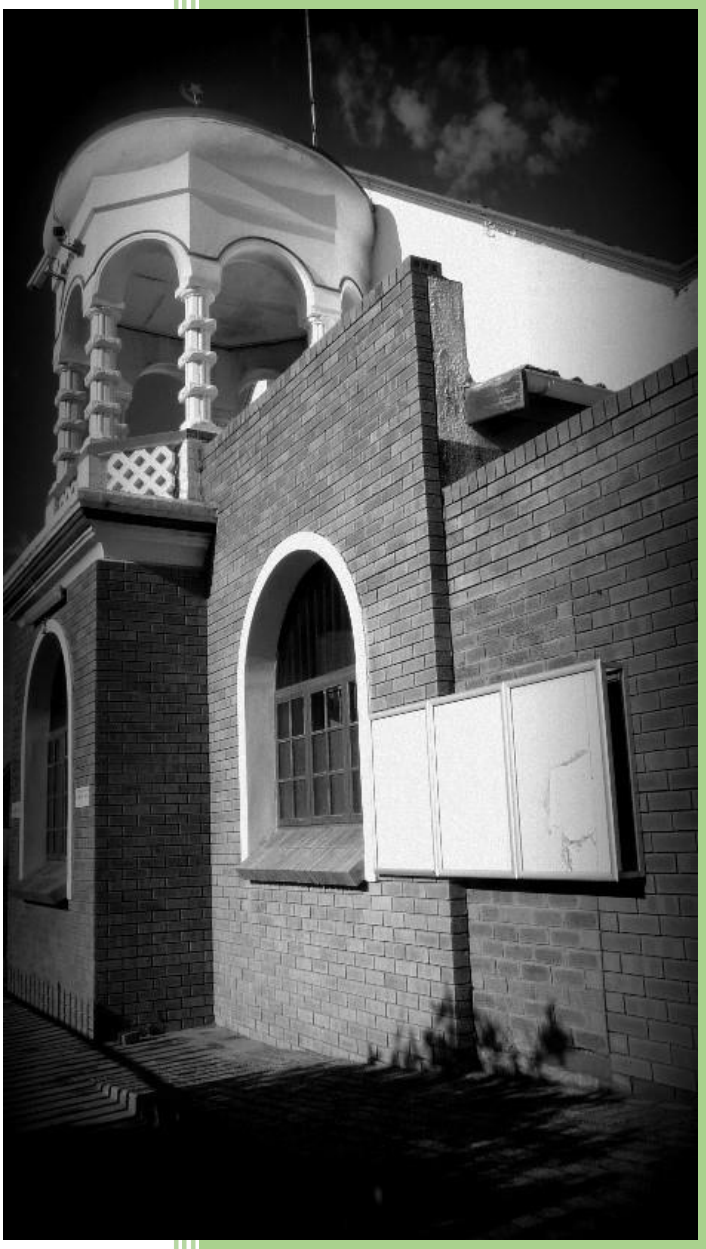
Product Requirements
Ticket Management



Goals

To build an easily accessible application to streamline management of ticket sales of GIS fundraising events.

- To monitor ticket sales
- To manage collection of payments
- To get a view of ticket sales per fund raising event
- To get a view of payments collected per fund raising event

Users

- Co-ordinator of ticket sales and fund collections
- Administrators of ticket sales and fund collections
- Co-ordinator should be able to perform administrator tasks

User Stories

- As a co-ordinator I want to capture a new fundraising event so that the resulting ticket sales can be tracked

- As an administrator I want to capture tickets allocated per person per event so that I can track payments received/outstanding

- As an administrator I want to capture payments received per person per event so that I can track payments received/outstanding

- As an administrator I want to capture tickets returned per person per event so that I can track un-sold tickets

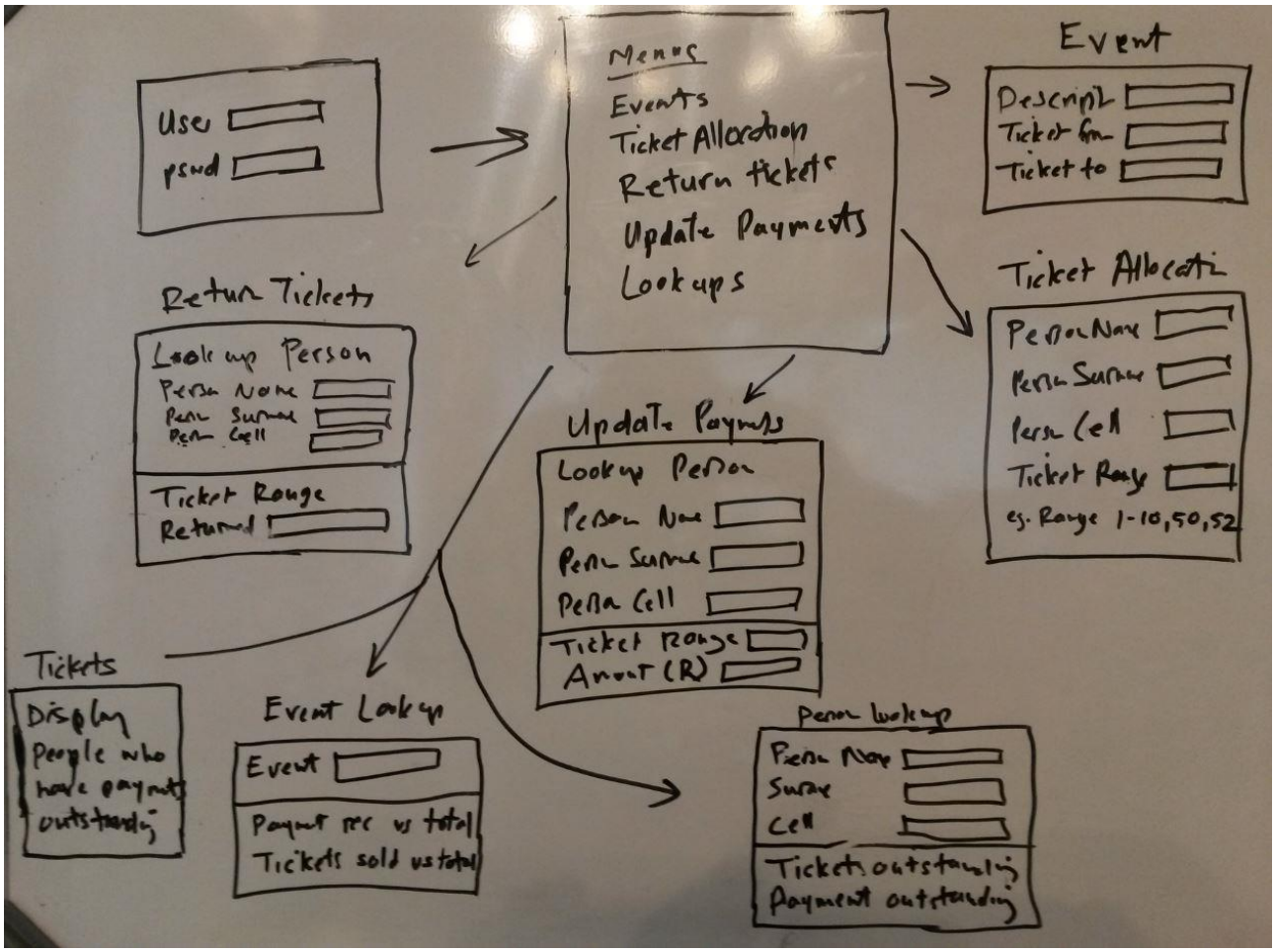
- As an administrator I want to look up a person so that I can check their outstanding payments

- As an administrator I want to look up a person so that I can check their outstanding tickets

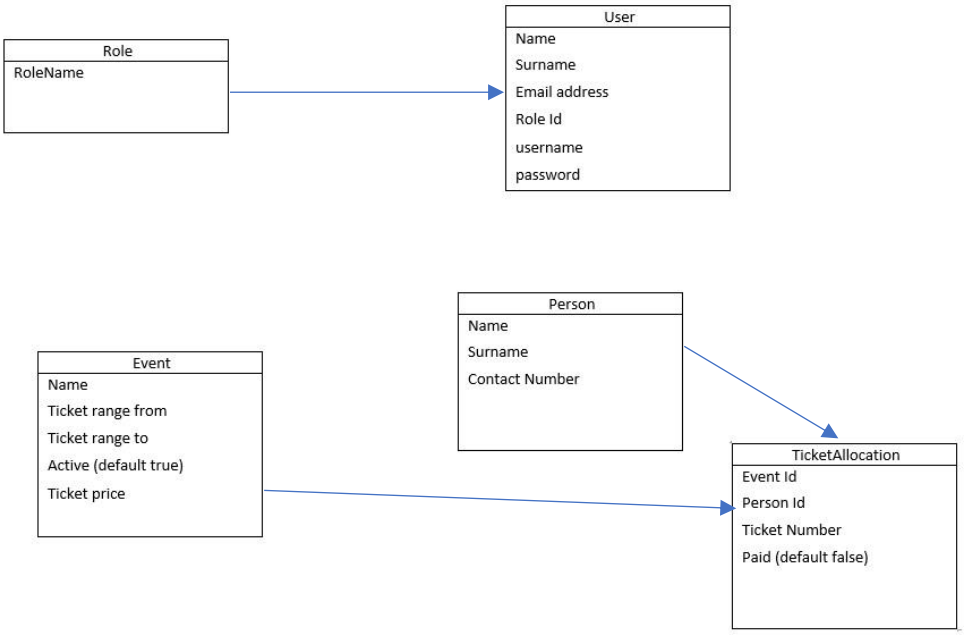
- As an administrator I want to query an event so that I can get a view of outstanding payments

- As an administrator I want to query an event so that I can get a view of un-sold tickets

Sitemap/Page Descriptions/Wireframes



Database design



Note:

- For each Person captured by a User, a **TicketAllocation** record will be inserted for each ticket in the range of tickets given/sold to the person. Eg, If Joe Soap was allocated a batch of tickets with range 100 – 110 then 10 ticket **TicketAllocation** will be created.
- **TicketNumber** should be within range on **Event** table.
- The **Paid** flag on **TicketAllocation** will be set to true for payments received.
- **User** table to be pre-populated with data supplied
- **Each Table to have is own ID column**

Non-Functional Requirements

- Android application
- Hosted mysql database
- Database updated in real time
- Back end persistence framework (eg hibernate) for CRUD code
- Back end exposed as webservice (eg. using jersey)
- Apk to be manually distributed

Images

Future Iterations

Out of scope:

- Audit Report generated on request via menu option
- Audit Report generated automatically emailed to each of users in User table
- Disable and Enable an Event
- Auditing

MVP Workaround

- Audit report manually exported directly from mysql db using SQL join