

Prepr Tagapp

This document provides an overview of the prepr tag app mobile application.

The application is meant to be able to scan QR codes and render associated content on a smart TV

The application is segregated into three main parts.

1. Mobile application
2. Prepr platform
3. Server responsible for rendering views on the TV and keeping a record of QR code key values

1. Mobile Application Views

The mobile application has 4 main views.

- Login
- Barcode Scanner
- History
- Dashboard

The login page is shown to new users of the application and acts

as an access portal. Authentication was handled by posting user input to the wordpress site, then inspecting the response.

The barcode scanner consists of the cordova camera API and a java zxing library included into the project.

The page consists of a button to scan barcodes, and selection options for different content types for the barcodes.

The history page lists the content of previously scanned codes and their scanned dates.

The dashboard page acts as an iframe to the online dashboard/content/control panel available at the Prepr main site for authenticated users.

2. Server Architecture

The mobile application relies on the existence of two external service servers.

1. Prepr main site/ prepr wordpress site
2. Barcode key-value database and API

1. The main prepr platform is used for user authentication and as a content source for the dashboard view.
2. The barcode key-value db and api stores corresponding data for keys stored in the barcodes and renders requested data.

It consists of three views.

- TV name registration page
- Empty template view
- Rendered template view

The user first goes to the TV name registration page at url say www.prepr.com/tv

On registering a TV name the empty template view is loaded.

The user enters their TV name and scans a QR code.

On a server API access event i.e. a POST to the server, the data associated with the scanned QR code key is rendered. This data can be anything from a URL (e.g. youtube videos) to text content.

On subsequent QR code scans the server destroys its current rendered template and renders the newly sent data.

3. Technical Overview

1. Prepr platform/main site

Unknown, most likely PHP, not much control over the source code.

2. API server, QR code key value DB

The server is written in plain javascript and uses Express for the server and APIs. The database is PostgreSQL and consists of two columns, key and value.

The code is currently available opensourced on Github on the accounts of Walter Miller and Kaustav Haldar. The tested deployment exists only on a heroku node owned by W. Miller.

3. Mobile application

The mobile application is written using an old version of Cordova. It is written in modern web technologies namely HTML, CSS and JS. It leverages:

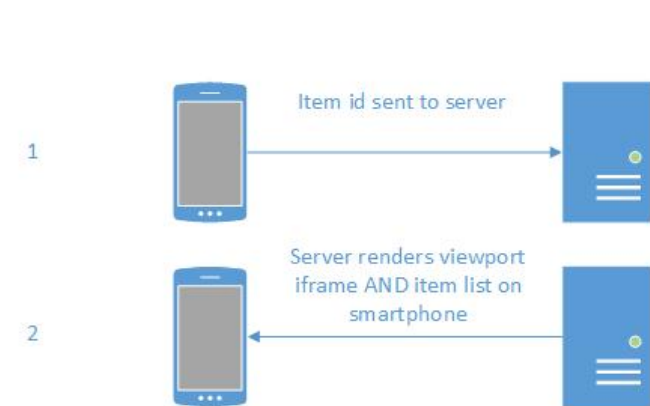
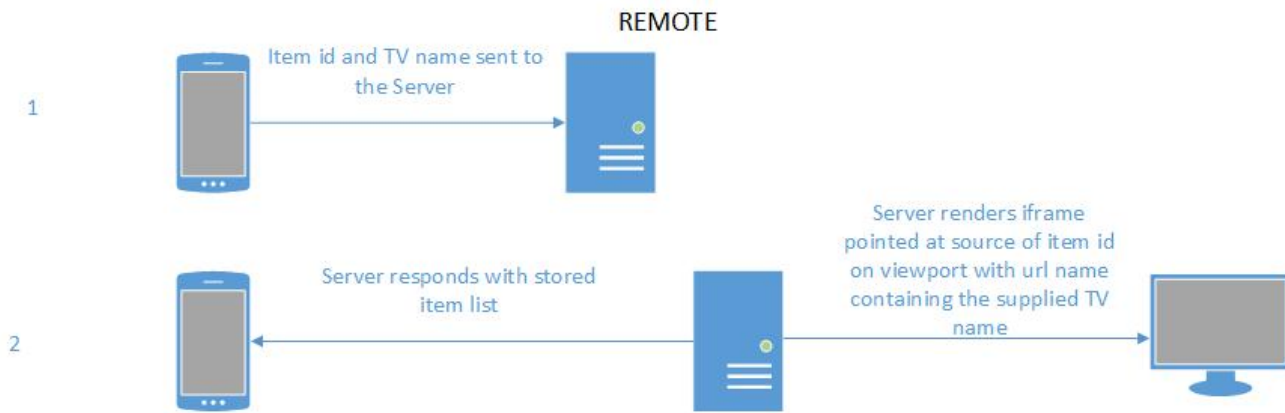
- Cordova barcodescanner API for scanning QR codes
- Localstorage for history
- jqmobile for UI and structure

4. Process

The process involved in using the app:

1. User opens app on phone
2. Logs in using their prepr credentials
3. The app checks that the user is registered with Prepr
4. User starts TV, navigates to the servers view page, e.g. www.prepr.com/tv
5. Selects/enters a TV name of their choosing
6. The TV browsers redirects to a empty template viewing page e.g. www.prepr.com/tv/example where example was the selected TV name
7. User enters the TV name on the mobile app
8. Scans a QR code
9. The mobile application sends a HTTP POST request with the key encoded in the QR code and the selected TV name to the server e.g {name: 'example', item: '1'} or in url form "www.prepr.com/tv/:tv_id/item/:item_id"
10. The server searches the DB for the received key and renders the associated content into the view. I.E www.prepr.com/tv/example would start playing a youtube video at this point if we assumed we have '1': 'someYoutubeUrl' in the DB
11. The mobile application creates an independent frame within pointing at the server view rendered on the TV though it can be configured to point to a different view or allow control of the TV view.

Data flow diagram



About the serverside software



The server basically holds a list of item id's and the representative content source

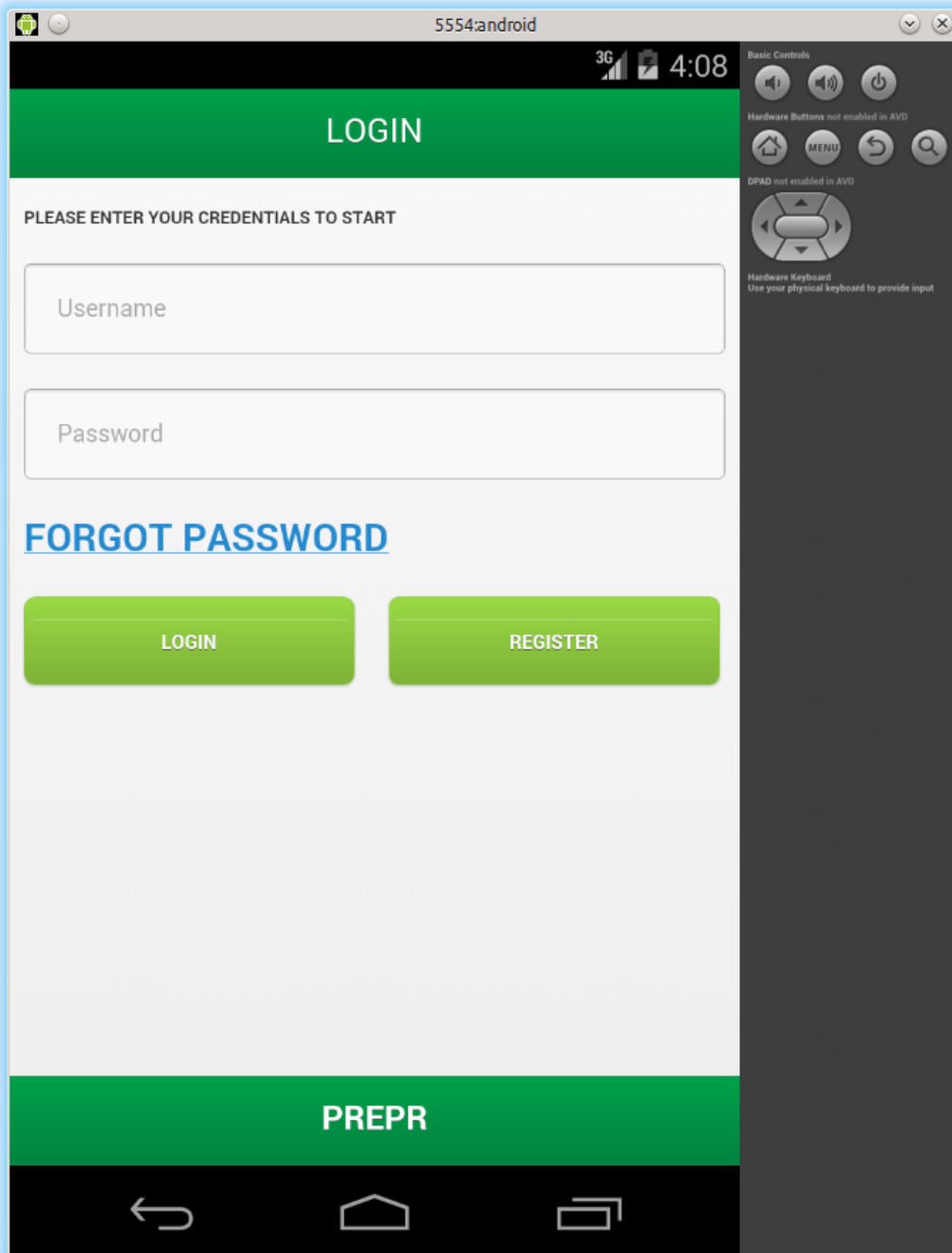
E.g
Item id
1
2

src
www.youtube.com&vid=1
www.youtube.com&vid=1

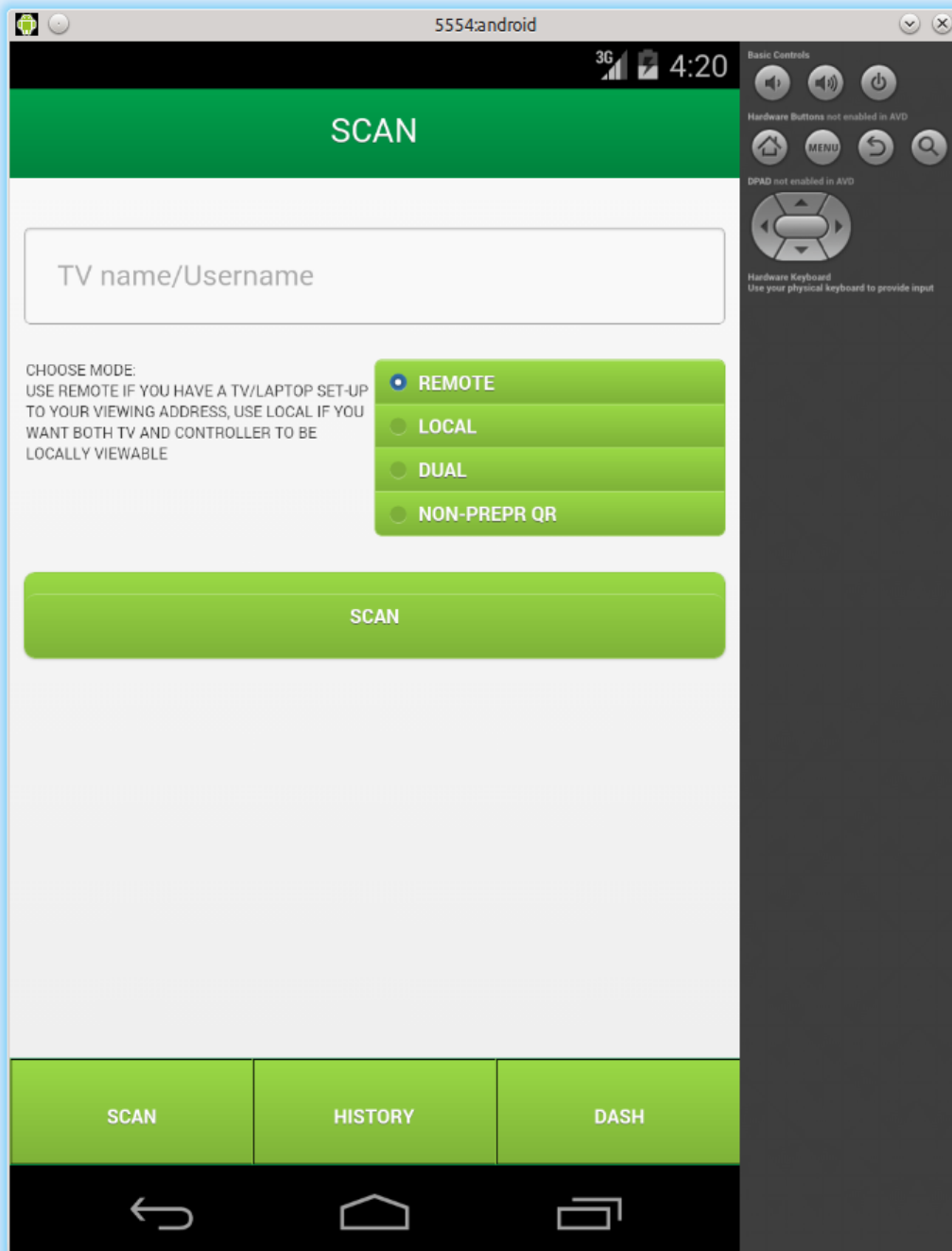
The servers response is to display the list, and render the content associated to the received id on a viewport with the supplied TV name

5. Images of the current mobile UI

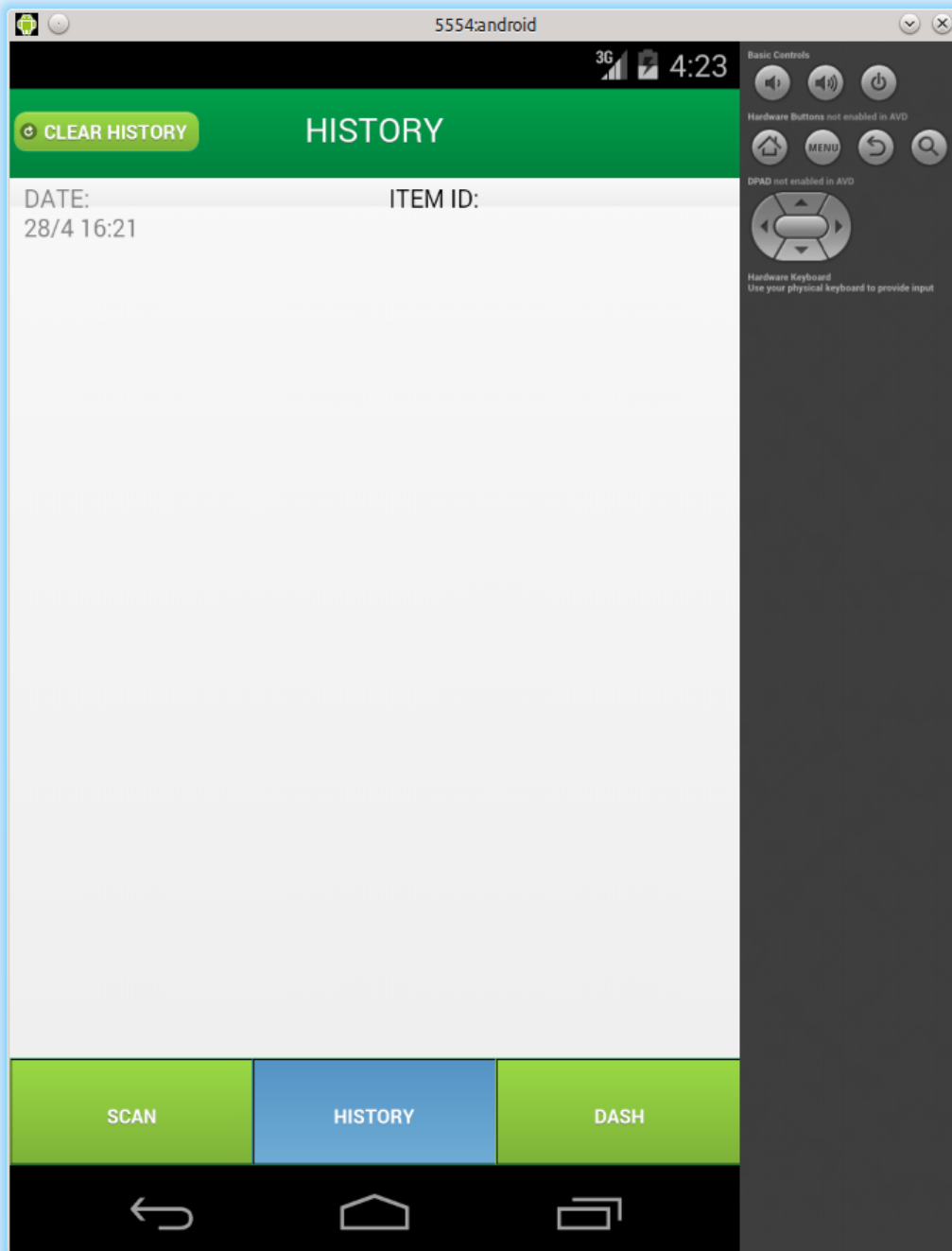
Login Page



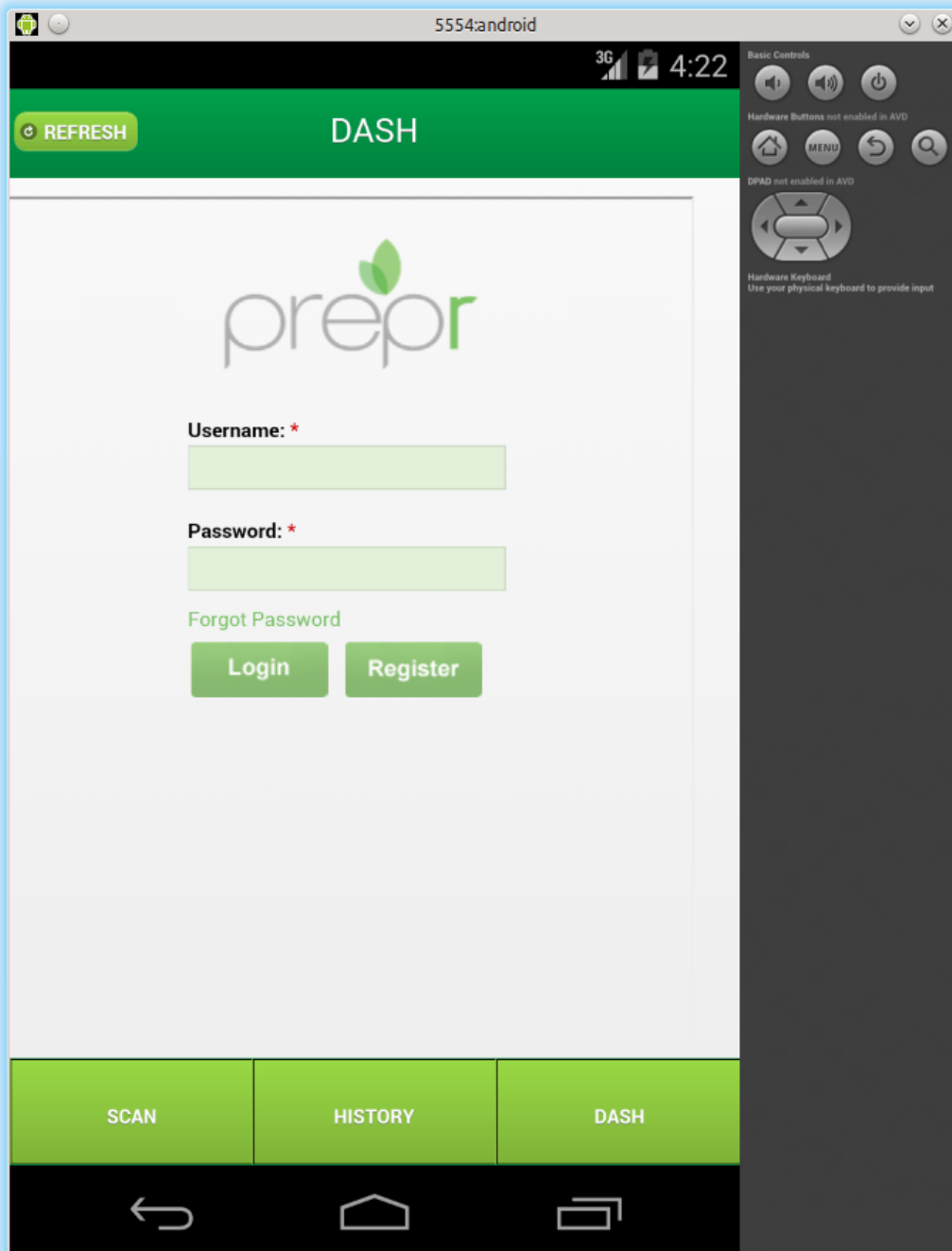
QR code scanning page and options



Scan history page



Dashboard login page



Logged in dashboard page

