UASApp: An Android App for Unallocated Space

{0}

Table of Contents

1 Introduction

This document contains the software design specification for the Unallocated Space Android application (UASApp). The purpose of this specification is to define how the UASApp software will be built and how it will interface with other UAS software and services. This document will also serve to allow anyone involved with UAS to either take over development, testing, or maintenance of the app at some future point. It may also serve as a useful resource to anyone wishing to recreate the app or some level of the app's functionality.

1.1 Unallocated Space

This Android app is intended for use by members and patrons of the local hackerspace, Unallocated Space. Unallocated Space is a technology-based community center, based out of a 1600+ SF space located in Severn, MD. This space is open to the public at all times and is available for people to come and collaborate on their projects. While we have interests ranging from electronics to woodworking, our primary focus tends to lie in Information Security. Our members offer a steady stream of talks and classes which are free and open to the public.

2 System Overview

The UASApp for Android will provide value to Unallocated Space members with the following features:

- Determine if the Hacker Space is open (as there is not a fixed schedule).
- Show the latest Camera Image.
- Alerts for new posts on the Unallocated Space Site and Twitter feed.
- Read and write to the Unallocated Space LED sign (members only).
- See who is at the space or is planning to show up.
- Manually or automatically check in at the space (members only).
- Register intent to check-in at the space at a given time (members only).

3 Design Considerations

This section describes many of the issues which needed to be addressed or resolved before attempting to devise a complete design solution.

3.1 Assumptions and Dependencies

The UASApp software design makes the assumption that a web-based API will be provided through which to access UAS information that is displayed to the UASApp user, and also through which information may also be submitted. The following services will require said UAS web-based API:

Services available to all:

- Check UAS status.
- Retrieve latest camera image.
- Read UAS sign.
- Check rollcall at the space for members checked-in or intending to check-in.

Services available to users who provide membership credentials:

- Verify user credentials.
- Write to UAS sign.
- Check-in at the space.
- Check-out at the space.
- Register intent to check-in at a given date/time.

3.2 General Constraints

- Minimum Version of Android: 2.2
- Optimal Version of Android: 4.2

4 VersionControl

The application source code will be kept under version control via GIT using the UAS GitHub repository which can be found at: https://github.com/Unallocated/UnallocatedApp-Android

6 Detailed System Design

6.1 Use Cases

The following use cases describe the intended actual usage for each feature of the UASApp.

6.1.1 UAS Status

A user upon starting the application will see a statement stating whether the space is open or not. Additionally, there will be an icon to visually indicate the open status of the space. The icon color will be GREEN if the space is open, RED if the space is closed, and BLACK if the network resource is unreachable. In the final implementation, there will be an additional status of YELLOW used to indicated that the space is not yet open, but that a keyholder has registered the intent to check-in and open the space later that day. Upon successfully checking the state of the space, the application will store this state for the background service in a SharedPreference Object.

6.1.2 UAS Status Alert

If the user has configured the application to alert the user when the space opens, the following will occur. If the space was previously closed, then a background service will check every thirty minutes (configurable by user?) to see if the space is open. If the space is open, then the background service will notify the user. If the space was previously open, then a background service will check every two hours (configurable by user) to see if the space is closed. If the space is closed, then the background service will remove any "open" notifications that are present. The time when the background service checks for updates will be referred to the scheduled refresh timer by other features.

6.1.3 UAS Wall Image

When the user accesses the home screen, the application will display the latest wall picture. This picture is pulled from cache if the network resource is unavailable. The user should be able to save, share, or view the picture in full screen. The picture will be cached to the file system, and a pointer will be stored as a SharedPreference object.

6.1.4 UAS Events Listing

When the user accesses the home screen, the application will display the any events listed on the UAS calendar for the current calendar day. The current day's events listing will be cached in a SharedPreference object.