Software Requirements for ARPost-its

Mobile Application for Agile Software Development using Augmented Reality

Author	Dhiana Deva Cavalcanti Rocha
	www.github.com/dhiana
Class	Software for Smartphones and Cloud Computing (EEL970)
Instructor	Sergio Barbosa Villas-Boas (sbVB)
	www.sbvb.com.br
Institution	Department of Electronics Engineering (DEL)
	Polytechnic School (Poli)
	Federal University of Rio de Janeiro (UFRJ)

Overview

AR Post-its is an Android application using augmented reality to display virtual objects on top of encoded post-its. A virtual progress bar can change size and color according to information available on a cloud-based web system.

Purpose

AR Post-its aims to help agile teams keeping an organized task board and effective stand-up meetings. It provides an intermediate solution to between paper-based and software-based agile task boards.

Target

Software developers, project managers, product managers and stakeholders of agile projects.

Scope

AR Post-its can be used as a general productivity application. Still, the early development will focus on the agile software development context.

Functional Requirements

The *User* can sign in using Google+ authentication.
The *User* can login in using Google+ authentication.
The *User* can create a new *Project*.
The *User* can choose an existing *Project*.
The *User* can create *Items* related to a *Project*.
The *User* can create *Tasks* related to an *Item*.
The *User* can see virtual progress bars displayed below Post-its with encoded frames.
The progress bar should communicate the progress and status of the *Item*.
The *User* can update the progress and status of the *Item*.
The *User* can see the list of *Tasks* related to an *Item* represented by the Post-it.

Software Architecture

Components

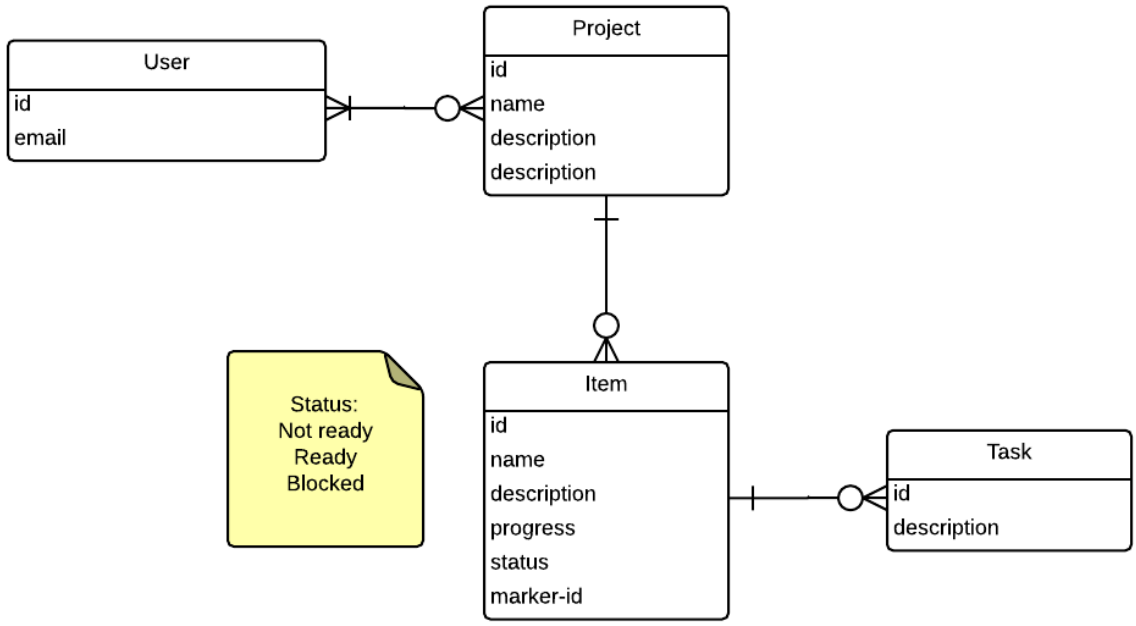
Mobile application

Users will view the augmented world by targeting their mobile camera on Post-its with encoded frames.
The mobile application uses Vuforia mobile vision platform for augmented reality and Retrofit as the RESTful client for cloud integration.

Cloud application

Users will be able to add, edit and delete *Projects*, *Items* and *Tasks* on a web application.
A RESTful API will be available so that the mobile application can request informations about these entities.
It will be developed in Python using Django web framework and Tastypie for RESTful API.

Data Model



User

Users will be team members (developers, managers…) and stakeholders of a project.
They will be uniquely identified by an email address. Google+ integration will be responsible for authentication. None of the applications will allow unauthenticated users.

Project

In the agile development context, a project has a physical task board where the items selected to be developed on an iteration are represented by Post-its.
A project involves a group of Users (team members and stakeholders).
Projects allows different teams working on different projects to use the same markers (0~511) without conflicting informations.

Item

Items are represented by Post-its on agile task boards.
The physical Post-its will be enriched by a virtual progress bar where it's length will be proportional to the item progress (0~100). Its color will communicate the item status:

- **Green**: Ready (to be developed)
- **Yellow**: Not Ready (waiting for requirements)
- **Red**: Blocked (waiting for the resolution of impediments)

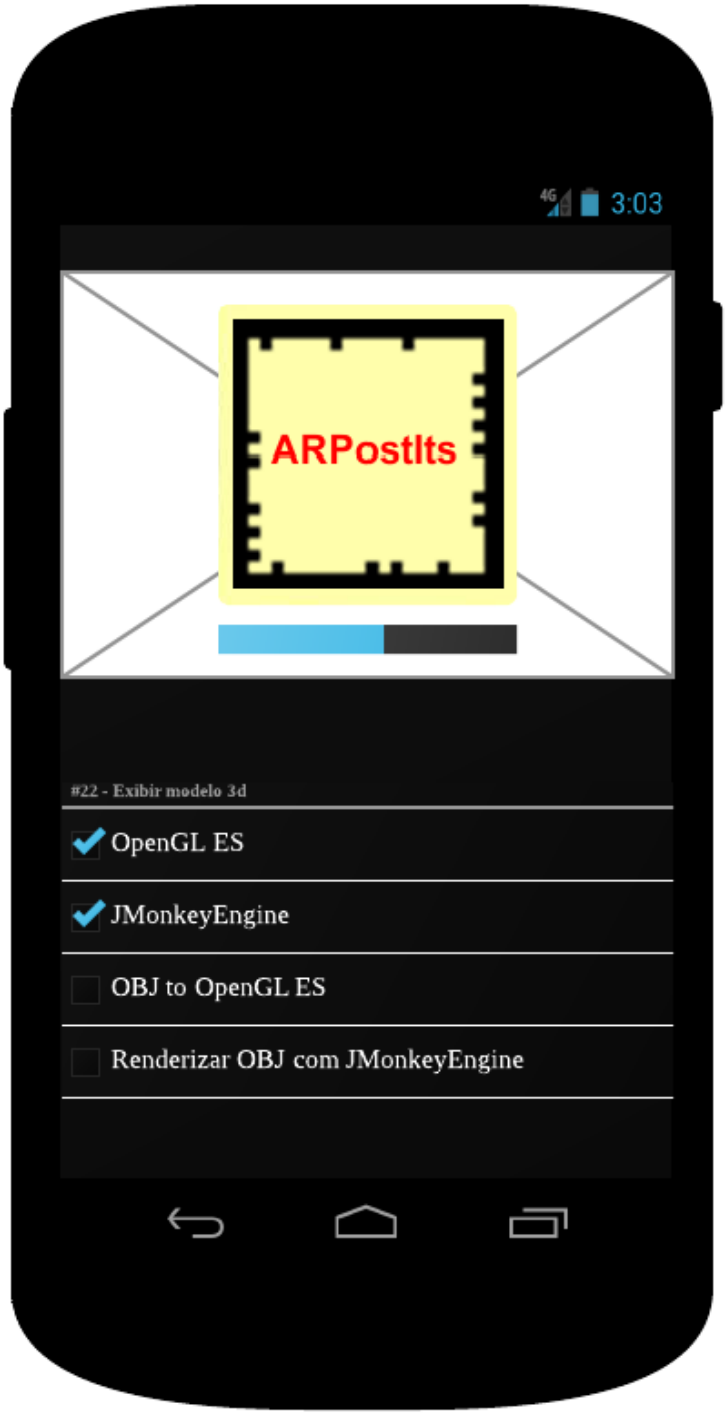
Task

Agile teams often stick Post-its on top of one another to aggregate more information about an item.
These extra informations can be organized as a task list that will be displayed on the mobile application.

Technical Requirements

The applications (mobile and web) must be open-source (MIT license) and available at Github.
The applications must be developed with Automated Testing and Continuous Integration.
The web application will be hosted on Heroku, a Platform-as-a-Service (PaaS).

User Interface



Assumptions

The user must print the encoded frame on Post-its.

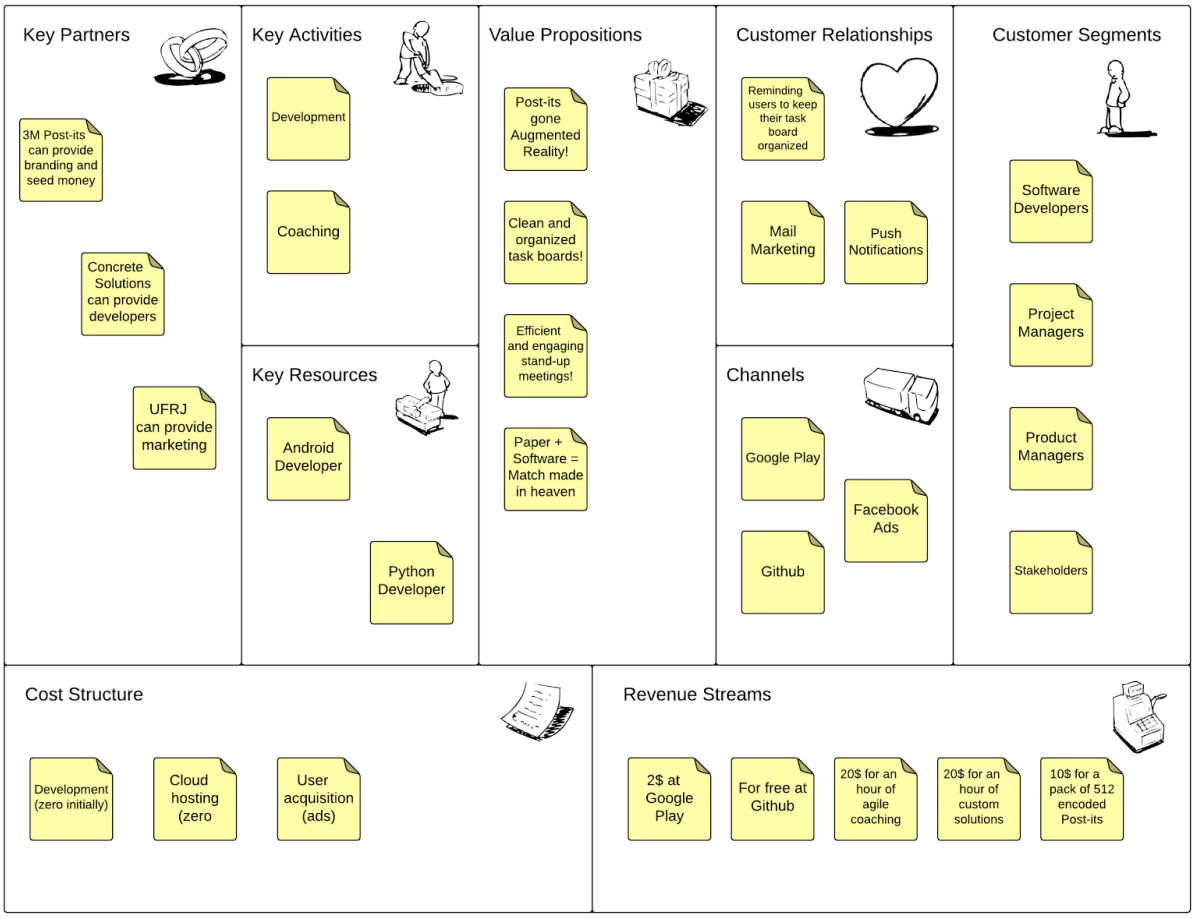
Constraints

The application must have a stable connection to the internet. This constraint is not a threat to adoption because the application is targeted to be used at organizations offices, where wifi is certainly available.

Business Model

The application will be available for download as a paid application at Google Play. The code will be available at Github as an open-source project. Those who don't want (or don't know how) to build and install the application from the open-source code can pay for the application at Google Play. Paying at Google Play will also allow receiving automatic updates. Initial price will be set to US$2,00 (typical productivity app).

The Business Model Canvas



Project Plan

Milestone #1

The requirements document must be done.
The project's website must be live and should contain relevant information about it.
The development environment should be ready.
The application must be available at Github.
The development environment should be able to run a "Hello World" application.
Automated Testing should be able to test if the "Hello World" text appears.
Continuous Integration should be able to build the "Hello World" application.

Milestone #2

The core functionality must be done: the mobile application must recognize Post-its with encoded frames and display a virtual progress bar according to information from the cloud.
The cloud application must be live.
The website and software package must provide templates for printing the encoded frames on Post-its.
The website must display a video of the working application and a tutorial for printing Post-its.
The development environment setup must be documented.
Unit tests should ensure the core functionality behaves as expected under controlled conditions.

Milestone #3

User authentication must be done.
Project creation and selection (on the mobile application) must be done.
Progress and status update (on the mobile application) must be done.
Virtual task list must be done.
Unit tests should ensure all functionality behaves as expected even at faulty conditions.
The application must be published to Google Play.
The website must display a link for the application page at Google Play.