**Projectbox**

An open source project-sharing application for the World Wide Web and mobile systems

****

**System Documentation**

Course: Software Engineering Project II (CICS 530)

Term: Winter-2, 2012

Submission date: 23rd April, 2012

Instructor: George Tsiknis

TA: Ricky Cheng

**Group Members:**

Matthew Robertson

Gayathiri Charathsandran

Ying Xu (Lillian)

Srikanth Ramu

Benjamin Bertka

Parvez Islam

Master of Software Systems (MSS)

Institute for Computing, Information and Cognitive Systems (ICICS)

University of British Columbia (UBC)**Table of content**

|  |  |  |
| --- | --- | --- |
| Topic |  | Page |
|  |  |  |
| 1.0 Summary |  | 3 |
| 2.0 Glossary |  | 3 |
| 3.0 Introduction |  | 4 |
| 4.0 System Architecture |  | 5 |
| 5.0 System Requirements |  | 6 |
| 6.0 System Configuration |  | 7 |
| 7.0 Performance requirements |  | 8 |
| 8.0 Use Cases |  | 9 |
| 9.0 User Interface Design |  | 10 |
| 10.0 Behaviour Diagram |  | 11 |
| 11.0 Database Design |  | 14 |
| 12.0 Mobile Platform Specification |  | 17 |
| 13.0 Authorisation and Authentication |  | 18 |
| 14.0 Overview of Tests |  | 19 |
| 15.0 Known bugs |  | 21 |
| 16.0 Lessons Learnt |  | 22 |
| 17.0 Conclusion |  | 23 |

# 1.0 Summary

*Projectbox* is a simple collaboration tool meant for both web and mobile platforms. It is easy to use and a user can have a project uploaded, shared and available for collaboration in minutes.

# 2.0 Glossary

1. Projectbox: This is the name of our system. The system can be used for hosting, sharing and contributing Open Source projects. There are contributors who help in the technical aspects of a project while project managers are the owner of a project
2. Web Platform: This pertains to the web interface which can be accessed using a browser from any desktop and laptop computer.
3. Mobile Platform: This pertains to the mobile interface which is meant for devices with limited computing resources such as iPhone, Android phones, Blackberry, etc.
4. RoR/Rails: Ruby on Rails MVC framework.
5. Project manager: A project manager in the context of our Projectbox system is the owner/creator of a project. A project manager may add/delete contributors to/from a project.
6. Contributor: A contributor is created when either invited to join a project or the project manager accepts a user’s request to join a project.
7. Administrator: The administrator of Projectbox has access to all projects and user information. An administrator reserves the right to create/modify/delete a project/user.

**3.0 Introduction**

Projectbox is a simple collaboration tool meant for both web and mobile platforms. It is an interactive and easy to use tool where user can have a project uploaded, shared and available for collaboration in minutes from anywhere. Project upload can be anything files, images or video clips to the project. It was built using Ruby on Rails MVC architecture and phone gap mobile framework. It is compatible with Android smart phones to access from anywhere.

This powerful tool can be used for collaborating professionals in a frame work, ease their communication among projects and easily handle separation between personnel’s owned projects and contributing projects. It is an easy tool for Project Managers to create and manage their projects. This tool can be used to recruit project contributors via email or can ask the existing ProjectBox users to contribute. For each user of this tool, who interacts either by PC or smart phone is provided with appropriate interface to access the ProjectBox.

Administrators of this ProjectBox can maintain the overall system and send announcement for users. Project managers can create, update and delete their own project and can decide whom to join the project.

Below is a list of our intended usage:

Organization who wanted to

* maintain their project management activities or
* provide an easy tool for their project managers and professionals to perform their task effective from anywhere or
* run a business to provide the service

Users of ProjectBox,

* Project managers
* Professional who wanted to contribute to the project
* Users of the projects resources created by project managers

**4.0 System Architecture**

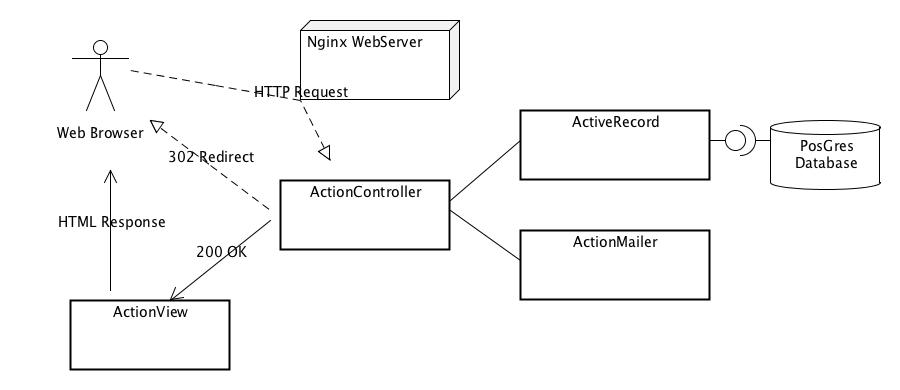


Figure 4-1: Rails architecture

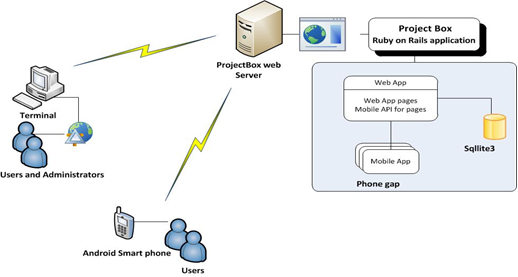


Figure 4-1: Interactions between project box servers and clients (desktop and mobile). Mobile architecture discussed more later.

**5.0 System requirements**

Administrator:

* Rails server
* SQLite3
* Phonegap Framework
* Android SDK
* JQuery and JQuery Mobile
* Domain name in the Internet

Project Managers / project contributors:

* Internet access
* Registration with service providers

**6.0 System Configuration**

The following instructions explain how to set up your development environment for Ruby on Rails on windows. If you want to use Linux or Mac (which I would recommend if you are comfortable with UNIX and the command line) then just wait for my next post which will describe the process on those systems.

Step 1: Download the Rails installer

* open your web browser and go to http://railsinstaller.org/
* click on the big button in the top left to "Download the kit"
* optionally, while it is downloading you can watch the video tutorial they have on the site.
* when the download completes double click the installer, say yes to
* any security warnings you may see, and accept the default installation directory (something like C:/RailsInstaller)
* once the install completes click finish, this will open up a terminal in which you are prompted to enter your name and then email

Confirming:

* open a command terminal
* enter the command 'ruby -v' (the response should begin with ruby 1.9.2p290)
* enter the command 'rails -v' (the response should be Rails 3.1.1)
* if either of the above are not correct something is wrong and you should stop and reply to this thread and I will try to help you get sorted out

Step 2: Install SVN

If you have another means of accessing SVN by all means you do not have to do it this way, it is just what I know.

Download the Tortoise SVN client installer from: http://tortoisesvn.net/downloads.html

When the download is complete double click the installer, and follow the installation process steps accepting all defaults

Step 3: Checkout the code:

* open up windows explorer and navigate to some directory that you would like the source code to live in right click any white space in the directory and choose 'SVN
* Checkout' from the context menu in the dialogue that shows up fill in the repository url as:
* https://cics530-430-cms.googlecode.com/svn/trunk/
* fill in the directory with a path on your system that you will be able to find again (remember this you will need it in step 4)
* make sure checkout depth is set to 'Fully Recursive' and 'HEAD revision' is selected
* click ok and be patient (it can take a second)
* when the checkout is finished click OK

Step 4: Installing the database and populating data

* Download and install SQLite3 from: http://www.sqlite.org/download.html
* Browse to cics530-430-cms/cms\_rails/ using the Rails CMD application
* type the following:

rake db:create:all

rake db:migrate

rake db:seed

Step 5: Run the code:

* open up a terminal and cd into the directory you checked out the code into in Step 3
* type the command 'dir' you should see the following entries: docs, cms\_rails
* cd into cms\_rails
* type the command 'rails s' this is an alias for 'rails server' and will start the rails development server running on port 3000
* on the first run you may need to approve some firewall settings
* open your web browser and navigate to http://localhost:3000
* you should see the rails 'Welcome aboard, You’re riding Ruby on Rails!' page

Step 6: Run Webrick server:

* if you made it this far you now have everything you need to begin rails development

**7.0 Performance Requirements**

This section specifies non-functional requirements such as:

* Response times: we would expect the system to respond to each request in 1 second
* Capacities (number of simultaneous users, database sizes, etc.): We have deployed the system on Amazon AWS servers which provides distributed services. This means our system can handle infinite number of requests simultaneously and the user database can grow infinitely.
* Reliability and availability: We have built the system with both reliability and availability in mind. The RoR system itself provides several options for this.
* Security: Rails as an MVC system provides built in security such as password encryption-algorithms, seeding and standard sever side security.
* Usability: We have successfully managed to design the system with high usability in mind. We have used twitter-bootstrap code to simplify the user experience. The theme has been incorporated from several popular websites (just the idea).

**8.0 Use Cases**

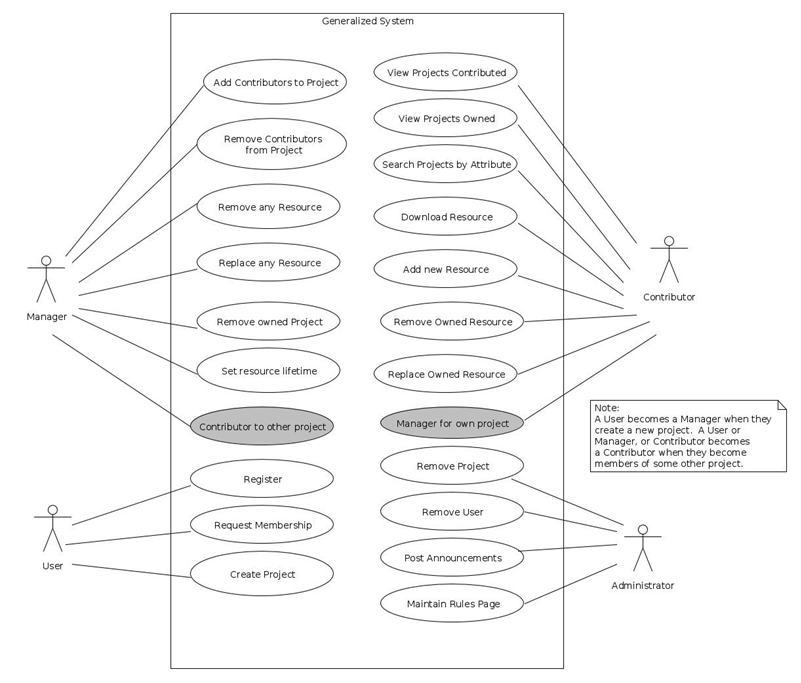


Figure 7-1: Generalised use case diagram

# 

# 9.0 User Interface Design

We have used the Pencil tool to draw UI mock-ups before we actually deployed the design. Below are some of the pre-sketches, pencil diagram and final look:

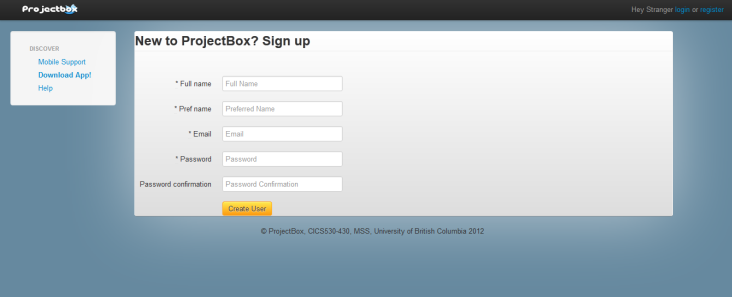
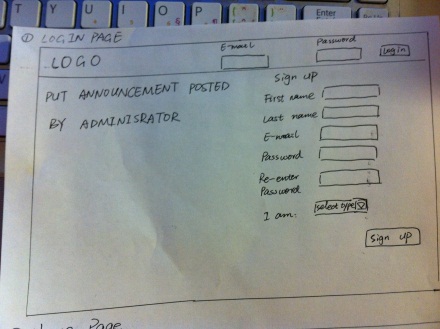
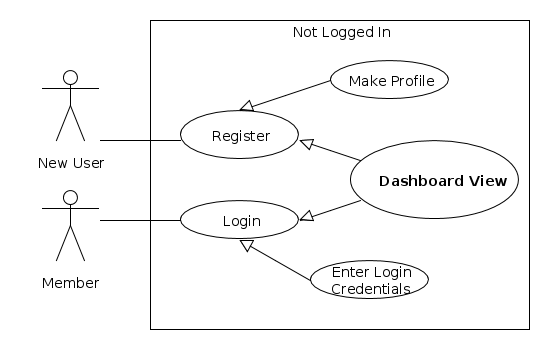
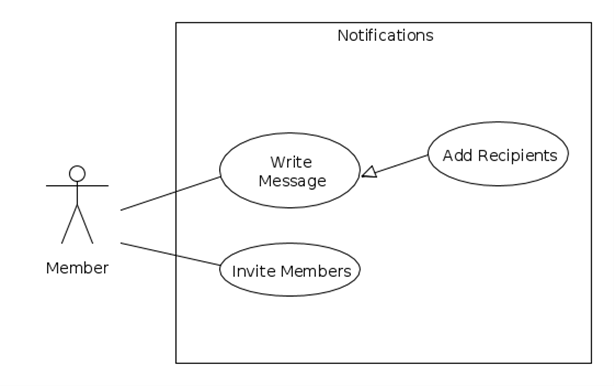
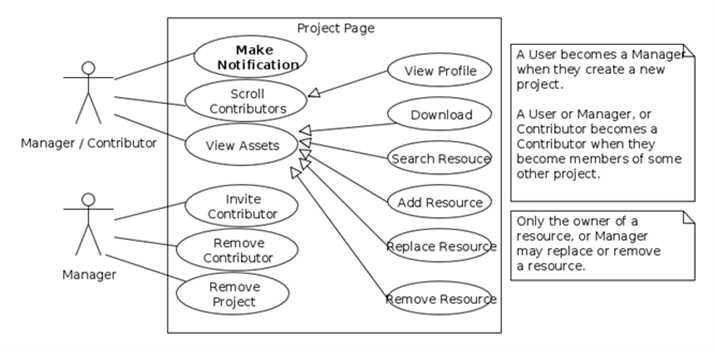
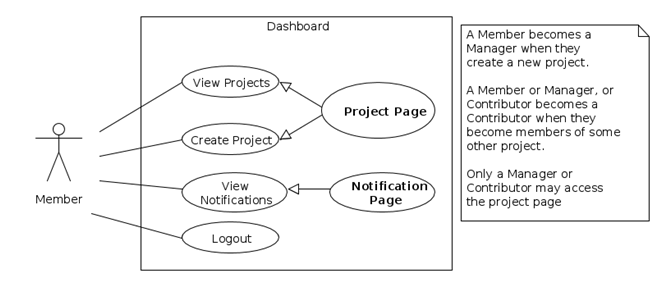
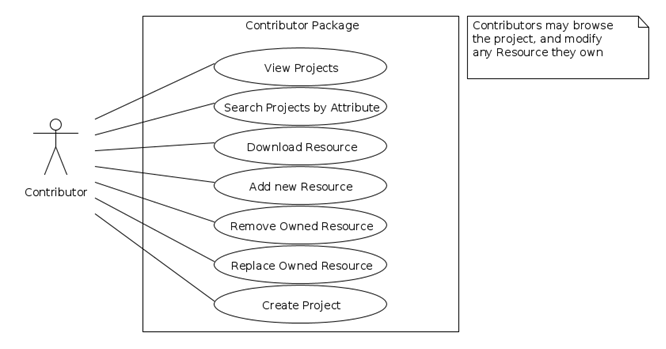
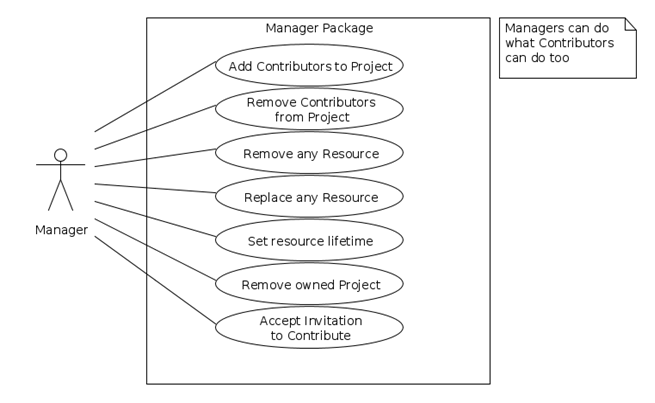
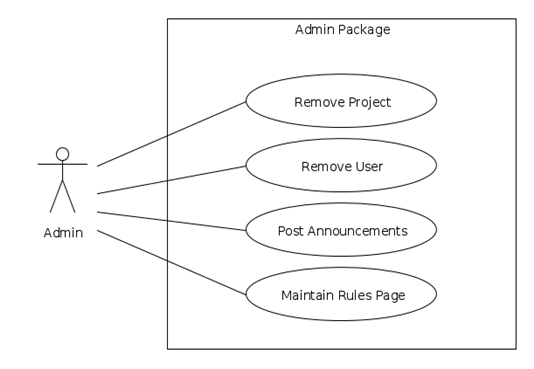
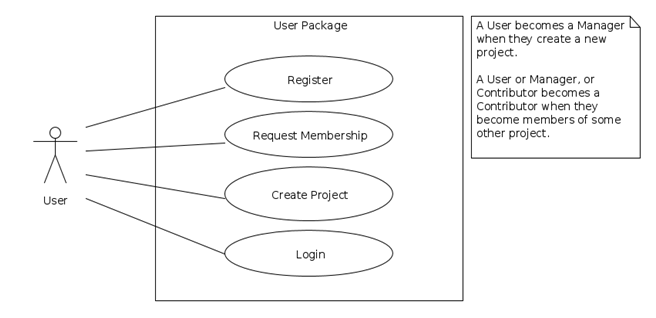
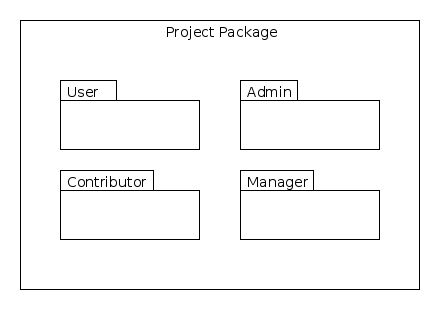


Figure 8-1: Progression of UI work

# 10.0 Behaviour Diagrams



# 11.0 Database Design

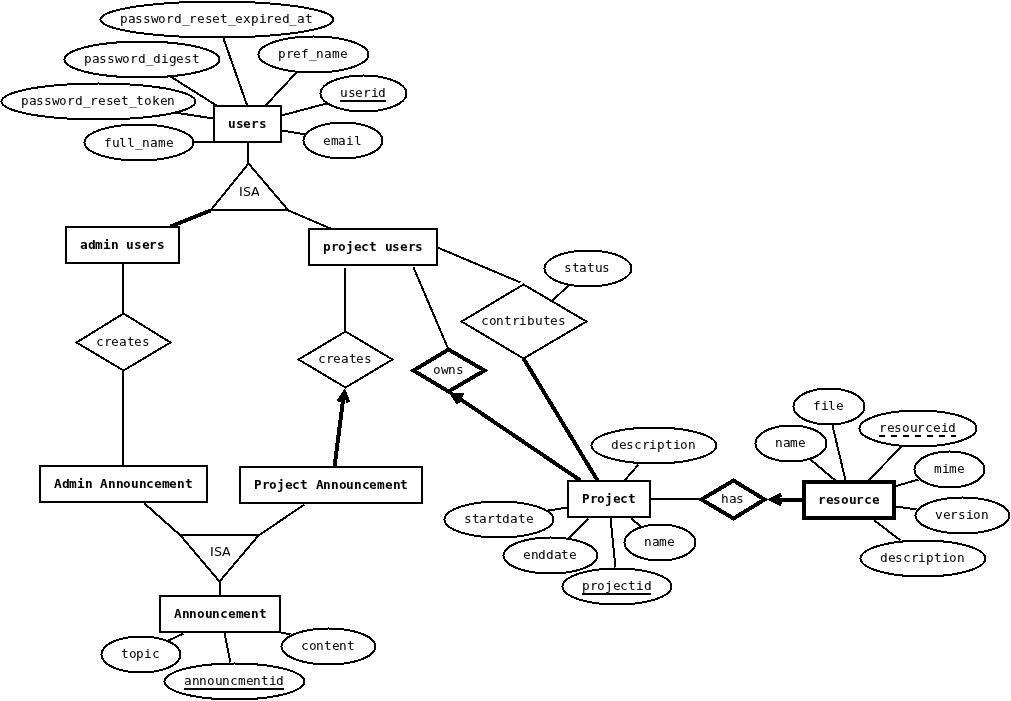


Figure 10-1: Entity-relationship diagram (see next section for detailed description)

**11.1 Primary tables**

We have used Ruby on Rails for the development with back end Sqllite3 database. This database has a convention on building the database. Particularly, each table has id, created\_at and updated\_at as default attributes to the tables. Hence we have used the convention when stating the tables below.

1. Users(userid, full\_name, pref\_name, email, admin, password\_digest, password\_reset\_token, password\_reset\_expired\_at)

Represents: entity set Users of the projects

Primary key: (userid)

Constraints: Users id should be unique, admin is true for administrators

1. Project(projectid , started\_at, ended\_at, name, description, ownerid)

Represents: entity set Project and relationship set owns

Primary key: (projectid)

Foreign key: (ownerid) references Project owner

Constraints: Project can be owned by a single user, but user can own multiple projects (project id is unique). Ownerid is not null.

1. Contributions (userid, projectid, status)

Represents: entity sets Users ,Project and relationship set Contributes

Primary key: (userid, projectid)

Foreign key: (userid) references Users, (projectid) references Project

Constraints: Project can have many contributors and contributor can work in many project. But user must request the project owner to contribute. By default user status 0. If user request owner for contribution, status 1 and when owner accept user to contribute status 2. Projectid is not null.

1. Resource(resourceid, projectid, name, version, description, file, mime)

Represents: entity sets Resource and Users and relationship set has

Primary key: (projectid, resourceid)

Foreign key: (projectid) references Project

Constraints: Project can own many resources. Projectid and resourceid is unique. Projectid is not null

1. Announcement(announcementid, userid, projectid, topic, content, kind)

Represents: entity sets Announcement

Primary key: (announcementid)

Foreign key: (userid) references Users, (projectid) references Project

Constraints: Announcement can be kind of Administrator or and project manager. Administrator can post announcements to all the users, but Project manager can post announcements to his contributors. Announcementid, projectid and userid is unique. Userid and projectid are not null, but projectid is 0 for admin users.

**11.2Tales and there dependency**

***Base Table***

* Users(userid, full\_name, presf\_name, email, admin, password\_digest, password\_reset\_token, password\_reset\_expired\_at)

***Dependent tables***

Hierarchical order of table, need to be entered

User

Announcement table

Project

Contributions

Resource table

Figure 10-2: Object Relational Model

**11.3 Functional Dependencies**

The following is a list of FD's for each table:

1. Users(userid, full\_name, pref\_name, email, admin, password\_digest, password\_reset\_token, password\_reset\_expired\_at)

* useridà full\_name, pref\_name, email, admin, password\_digest, password\_reset\_token, password\_reset\_expired\_at

1. Contributions (userid, projectid, status)

* userid, projectidà status
* projectid à userid
* userid à projectid
* userid à status

1. Project(projectid , started\_at, ended\_at, name, description, ownerid)

* projectid à started\_at, ended\_at, name, description
* projectid à ownerid

1. Resource(resourceid, projectid, name, version, description, file, mime)

* resourceid, projectid à name, version, description, file, mime
* resourceid à name, version, description, file, mime
* projectid à resourceid

5. Announcement(announcementid, userid, projectid, topic, content, kind)

* announcementid à topic, content, kind, userid, projectid
* announcementid, userid, projectid à topic, content, kind
* announcementid, userid, projectid à kind
* kind à topic, content, projectid

**11.4 Normalisation**

All the tables are in BCNF.

**12.0 Mobile Platform Specifications**

1. The mobile app is built over *PhoneGap* framework.
2. Supported features:
   1. Users can login to their account
   2. View their projects and Select a project
   3. Take a picture
   4. Choose a picture and upload it to their project
3. The app is tested in following configurations
   1. Android version - 2.3.4
   2. Device Model - HTC-Z710a

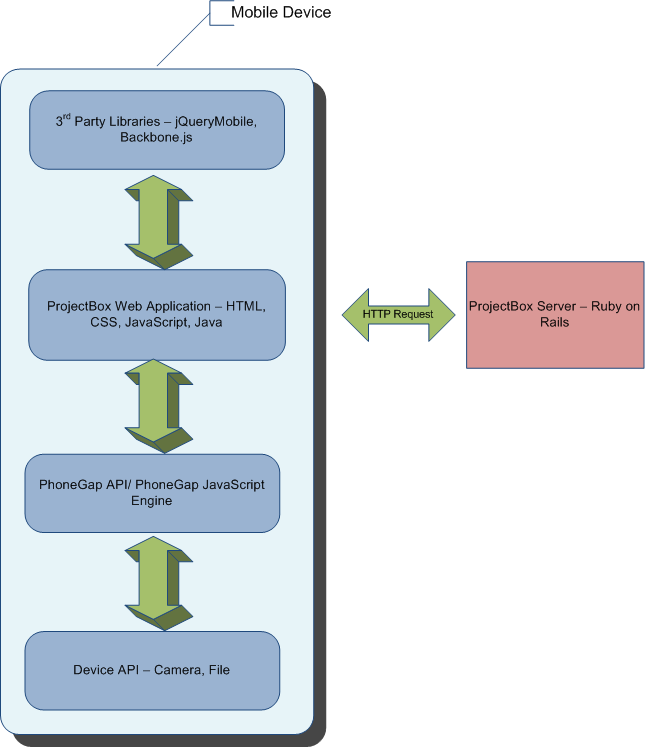


Figure 11-1: Interaction diagram of various layers in the mobile platform with rails server

**13.0 Authorisation and Authentication**

**13.1 Authentication System**

ProjectBox's authentication system is based on has\_secure\_password (ActiveModel::SecurePassword) of Rails 3.1. It offers following authentication functionality:

1. Makes attr\_accessor for password field
2. Provides confirmation validator of password
3. It hashes the password and stores it in the password\_digest field

**13.2 Encryption**

The system uses bcrypt-ruby hashing algorithm and it is a irreversible hashing mechanism.

**13.3 Authentication Process**

While authenticating the user, we use authenticate() function which takes unencrypted password. The password is hashed in runtime and compared with the password\_digest. If matches, the user is allowed else rejected.

**14.0 Overview of Tests**

Below is a list of Rspec tests (rspec spec --format documentation > ~/tests.txt):

AnnouncementsController

index

renders index template (FAILED - 1)

loads all the announcements (FAILED - 2)

show

renders show template (FAILED - 3)

finds and assigns the announcement (FAILED - 4)

new

creates a new announcement and assigns is (FAILED - 5)

new action should render new template (FAILED - 6)

create

renders new template when model is invalid (FAILED - 7)

redirect when model is valid (FAILED - 8)

edit

should render edit template (FAILED - 9)

update

should render edit template when model is invalid (FAILED - 10)

should redirect when model is valid (FAILED - 11)

destroy

destroys the model (FAILED - 12)

redirects to index action (FAILED - 13)

ProjectsController

index

renders index template

show

should render show template (FAILED - 14)

finds and assigns the project (FAILED - 15)

assigns contributors to project (FAILED - 16)

assigns resources to project (FAILED - 17)

new

creates a new project and assigns is (FAILED - 18)

new action should render new template (FAILED - 19)

create

should render new template when model is invalid (FAILED - 20)

should redirect when model is valid (FAILED - 21)

edit

should render edit template (FAILED - 22)

update

update action should render edit template when model is invalid (FAILED - 23)

update action should redirect when model is valid (FAILED - 24)

destroy

destroys the model (FAILED - 25)

redirects to the index action (FAILED - 26)

ResourcesController

index

renders index template (FAILED - 27)

loads all the resources (FAILED - 28)

show

renders show template (FAILED - 29)

finds and assigns the resource (FAILED - 30)

new

creates a new resource and assigns is (FAILED - 31)

new action should render new template (FAILED - 32)

create

should render new template when model is invalid (FAILED - 33)

should redirect when model is valid (FAILED - 34)

edit

should render edit template (FAILED - 35)

update

should render edit template when model is invalid (FAILED - 36)

should redirect when model is valid (FAILED - 37)

destroy

destroy the model (FAILED - 38)

redirects to the index action (FAILED - 39)

SessionsController

new

renders the new template if not logged in

redirects user if already logged in (FAILED - 40)

create

finds their user by email (FAILED - 41)

authenticates the user (FAILED - 42)

successful authentication

sets the users id in the session (FAILED - 43)

redirects to the users home page (FAILED - 44)

unsuccessful authentication

does not set the session id (FAILED - 45)

renders the new template (FAILED - 46)

destroy

it destroys the session token

redirects to the root url

UsersController

index

renders index template

loads all the users (FAILED - 47)

show

renders show template (FAILED - 48)

finds and assigns the user (FAILED - 49)

assigns the users managed projects (FAILED - 50)

assigns the users contributing projects (FAILED - 51)

new

creates a new user and assigns is (FAILED - 52)

new action should render new template (FAILED - 53)

create

create action should render new template when model is invalid (FAILED - 54)

create action should redirect when model is valid (FAILED - 55)

create action should set the sessions when model is valid (FAILED - 56)

edit

renders edit template (FAILED - 57)

update

update action should render edit template when model is invalid (FAILED - 58)

update action should redirect when model is valid (FAILED - 59)

destroy

destroys the model (FAILED - 60)

redirects to the index action (FAILED - 61)

MyResourcesHelper

add some examples to (or delete) C:/Users/Parvez/cics530-430-cms/cms\_rails/spec/helpers/my\_resources\_helper\_spec.rb (PENDING: No reason given)

PasswordResetsHelper

add some examples to (or delete) C:/Users/Parvez/cics530-430-cms/cms\_rails/spec/helpers/password\_resets\_helper\_spec.rb (PENDING: No reason given)

UserMailer

add some examples to (or delete) C:/Users/Parvez/cics530-430-cms/cms\_rails/spec/mailers/user\_mailer\_spec.rb (PENDING: No reason given)

Finished in 5.13 seconds

212 examples, 134 failures, 73 pending

Failed examples:

**15.0 Known bugs**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Index # | Bug type | Priority | Description | Status | Date found |
| 1 | Functional | Medium | help on explorer doesn't work | Resolved | 22-04-2012 |
| 2 | Functional/UI | Medium | if you click resource on explorer manu, nothing happens | Fixing | 22-04-2012 |
| 3 | Functional | High | cannot use forget password function | Resolved | 22-04-2012 |
| 4 | Functional | High | new user cannot login again using the registered email and password | Resolved | 22-04-2012 |
| 5 | Functional/Extra | Low | cannot send the invite letter to users | Unresolved | 22-04-2012 |
| 6 | Functional/Extra | Low | do we have advanced search? | Compensated | 22-04-2012 |
| 7 | Non-functional | Medium | Each action takes too long to execute | Resolved | 22-04-2012 |
| 8 | Functional | High | for admin sytem, admin cannot view, edit the information due the error message like this:  super: no superclass method `buttons' for #<ActiveAdmin::FormBuilder:0xb95d040> | Resolved |  |

**16.0 Lessons learned**

1. Rails is a fantastic platform for building applications quickly and efficiently
2. Rails in combination with Agile Scrum approach (which we roughly followed) to testing is a very good way to get applications up and running in a small time frame
3. We learnt about test driven development using *Rspec*
4. We added distributed application service using *Amazon AWS*

**17.0 Conclusion**

Projectbox is a promising application given the requirements and feature-set set by the course instructor. It gives a seamless user experience and lucrative UI. However, there are may be a few bugs in the system which can be fixed and other features can be refined given enough time and resources. Overall, we all enjoyed working on this project as a collaborative unit.