

# **Intentional Delivery Of Mail (IDM-UpSnap)**

*Project report submitted in partial fulfillment of the requirement for the degree of*

**BACHELOR OF TECHNOLOGY**

**IN**

**Computer Science and Engineering**

**By**

**Swapnil Khanna (141346)**

Under the supervision of

Mr. Mrityunjaya Shukla



**JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT,**

**SOLAN, H.P., INDIA**

May 2018

## DECLARATION

I hereby declare that the work represented in this report entitled “**Intentional Delivery of Mail**” in partial fulfillment of the requirements for the award of the degree of **Bachelor in Technology in computer Science and Engineering** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from August 2017 to May 2018 under the supervision of **Mr. Mrityunjaya Shukla**

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

*(Student Signature)*

*Swapnil Khanna(141346)*

*This is to certify that the above statement made by the candidate is true to the best of my knowledge.*

*(Supervisor Signature)*

*Mrityunajaya Shukla*

*ATL / PM*

## **ACKNOWLEDGEMENT**

The satisfaction and euphoria that accompany the completion of the project would be incomplete without the mention of the people who made it possible.

I would like to take the opportunity to thank and express our deep sense of gratitude to our faculty mentor Mr. Mrityunjaya Shukla for providing their valuable guidance at all the stages of the study, their advice, constructive suggestions, positive and supportive attitude and continuous encouragement, without which it would have not been possible to complete the project. The blessing, help and guidance given by him from time to time shall carry us a long way in the journey of life in which we are about to embark.

I am grateful for their cooperation during the period of the project. I hope that i can build upon the experience and knowledge that i have gained and make a remarkable mark in the future.

# CONTENTS

Certificate.....	(i)
Declaration.....	(ii)
Acknowledgement.....	(iii)
List of Figures.....	(v)
List of Tables.....	(vi)
Abstract.....	(vii)

Chapter-1	Introduction.....	1
Chapter-2	Literary View.....	11
Chapter-3	System Development.....	13
Chapter-4	Implementation And Coding.....	25
Chapter-5	Testing.....	35
Chapter-6	Conclusion.....	41

References.....	45
-----------------	----

## List of Figures

S. No	Figure No.	Title
1	3.1	Tables Upsnap
2	3.2	Address Table
3	3.3	Advertiser Table
4	3.4	Agency Table
5	3.5	Caller Analytics Table
6	3.6	Campaign Daily Report
7	3.7	Campaign Table
8	4.1	MVC Structure
9	4.2	Spring Directory Structure
10	4.3	Domain Classes
11	4.4	Services
12	4.5	Views
13	4.6	Controllers

## List of Tables

<b>S. No</b>	<b>Table No.</b>	<b>Name</b>
1	3.1	Hardware Environment
2	3.2	Software Environment

## **Abstract**

To the New, is a digital technology company providing end to end product development services. The company builds web and mobile applications that enables digital transformation for business. My work also revolves around the same. The project i've been working on works for the company client in Canada, US. The work is to develop the web applications for the client's website. The team working for the same project, that is handling back end of the website, [www.idm.upsnap.com](http://www.idm.upsnap.com), for Lance Brown. Technology used for working on the project is Spring Framework, using JAVA JQUERY.

**Keywords:** technology, back-end work, client, website

# CHAPTER-1: INTRODUCTION

## 1.1 Introduction

Upsnap is an Marketing company for different companies in Canada.

Objective of UpSnap is to effectively reach new visitors to websites signed to its products. In this regard, we have built an Intentional Direct Mailing Platform(IDM) that will identify a new user coming to a product (signed for ad campaign with UpSnap) website, and send him a postcard, at his address, for the product by fetching the address of the new user.

## 1.2 Problem Statement

There are myriad number of new visitors to a website. Let us suppose that Adidas is a client of UpSnap. If a new user visits adidas website, he should be receiving a postcard from adidas. Technically it will be sent after our project has identified that the user is new and it's visited url will be the deciding factor behind sending the postcard content.

### Client Requirements:

- Client should be registered as an advertiser with UpSnap
- Client should attach the javascript chunk provided to him on his target websites
- He should be able to view details of mails sent and his expenditure on the platform.
- He should not be able to create any new campaign by his account. Write permissions will only be for admin or super admin
- The client on logging in should be shown the a datewise daily graph of total eligible visitors to his specific ad campaigns and total mails sent for the day.
- Depending on the agency information, the logo images and theme of the console will change.
- The client can fetch date-wise information for his ad campaigns using date filter put on pages.



### 1.3 Objectives

Create Pixel javascript to attach to websites, which will call UpSnap related servers with relevant data

Store relevant data in database for analytics

Send information to third party related to postcards that are to be sent to new website visitors

#### **From Upsnap's point of view:**

- Able to record visit details of users (time of visiting a website, address indicator, number of time of visits in a day etc)
- Central repository of all kinds of transactions happening on a campaign, related to its updated budget, running status, target urls etc
- Daily, monthly and weekly reports to keep all stakeholders updated about the product performance.

#### **From Administrator's point of view:**

- Ability to create and update new users/advertisers/agencies
- Ability to assign access roles to users created (Admin, Super Admin, Agency admin, advertiser admin etc)
- Admin can create ,start, pause and end any of the campaigns

## **1.4 Methodology**

We will store the hits asynchronously in a nosql database

Will process the entries in nosql periodically to assess which users are to be targeted for sending postcards

Will update the counts of mails sent to users

Will maintain campaigns for these postcards with their start and end date

## **CHAPTER-2: LITERATURE SURVEY**

This chapter reviews the literature on Direct Marketing in the 21<sup>st</sup> era. Direct marketing in today's scenario for organizations to develop and maintain strong customer relationships is becoming a key strategy. This method targets specific customers with personalised advertising and promotional campaigns are mainly targeted in order to help organisations increase campaign responses and to get a higher return on their investments. This project focuses on the organisational and managerial plans of the direct marketing process and aids in the investigation of the stages, activities and technologies required to effectively execute direct marketing.

Organizations operate in markets which are highly competitive and constitute of a volatile environment, where purchasing behaviour of the customers keeps changing constantly and poses an utmost difficulty in prediction. In such an era, direct marketing has made its way to enhance promotion campaigns as well as develop strong relationships with customers.

My project focuses on the UpSnap's new UpSnap Direct website, which enables advertisers to sign up for intent-based direct mail campaigns using UpSnap's proprietary Intentional Direct Mail ("IDM") service.

Using Intentional Direct Mail technology, advertisers would be able to expand their audience by the key means of identifying their previously anonymous site visitors and then by sending a personalized postcard to their residence within days of their site visit. This somehow aims at joining merchants and buyers together to impart a highly successful business strategy. This has relatively added an extra layer of suit in the company's existing advertising software. There are numerous smart moves which are utilised by marketers in order to ward off their websites from unusual traffic. The Intentional Direct Mail completely fits with the advertising suite of products.

The main aspect of the IDM products is to amplify the efforts needed to drive the conversion with a physical mail for the customers to follow up on their initial expressed content and intent along with the increase in the number of sales.

To be able to leverage the identity dataset accounting as deterministic in the market, it has aided in identifying and qualifying non-converting prospects which could be derived once a person visits a particular site.

My objective of this projects accounts for the covering of the complete scope to engineer a product for the transformation of the businesses digitally. This review

Postcard Builder is one astonishing feature capable of creating a postcard which could be sent to you for no additional costs. This helps people to verify the quality of the product before thinking of investing in the company. This mailing service helps to handle the entire process to create and send mailers. A customized look for the mailers can also be designed or uploading of the existing designs on the postcards could also be done. Targeted mailing lists is one aspect that PostcardBuilder can provide you with which can be chosen from a residential or a business mailing list. Key customers based on the variety of information the person is provided with can be targeted. In a summarized view it could be stated that it is a sum of providence of mailing lists with assistance to designs and the complete assembly of all mailers.

The project IDM can be reviewed as- At first, the customer pays a vist to the website where he is captured by a pixel intended to retarget on the website. The available information is captured by the pixel which is then used to identify the user's home address.

Secondly, within a span of 24 hours, a customized direct mail postcard is headed to the home of the customers which offers a special giving for paying the visit to the site.

Thirdly, being the last step, the customer enjoys redemption of the special offer offered to him on the custom direct mail postcard.

Intentional Direct Mail has accessorised itself to be a tangible and an effective engagement tool which can now be spread and reached to the customers to help them improvise their sales and increase their profits.

## CHAPTER-3: SYSTEM DEVELOPMENT

### 3.1 Hardware, Software requirements:

#### 3.1.1 Hardware Requirements

The hardware environment in which project is developed is as follows :-

Hardware	Specification
CPU	Intel Core i7
RAM	16 GB
HARD DISK	1 TB

*Table 3.1 : Hardware Environment*

#### 3.1.2 Software Requirements

The server that we are using to host our application is “Heroku cloud application platform” which provides support for Agile deployments for Grails applications. The basic software support needed for the application is given in the table below.

	Software	Version
Operating System	Linux	Ubuntu 10.04.2
Web Application Framework	Spring	4.2
Database	MySQL	5.5
IDE	JetBrains IntelliJ Idea	12.0
Web Browser	Google Chrome, Mozilla Firefox	
Web browser plug-ins	Firebug, Awesome screenshot	
Version Control System	Git	

*Table. 3.2 : Software Environment*

## **3.2 Detailed System Specification:-**

### **3.2.1 Login/Authentication:**

The first step to access the application is to create a user account on its website. Only a registered user can add,edit the campaigns This process also allows a user(ADMIN) to sign up for the application if they are not already registered, then they can login with their credentials.

**Input:** The input to this process is user id and password.

**Output:** This process gives access to the application if the user id and password entered by the user are correct, otherwise it flashes an error and prompts the user that the credentials are wrong.

### **3.2.2 Create campaigns , advertiser , agency :**

After the user has successfully logged in, the user will have the ability to create a new campaign or he can edit to an existing campaign., when a user creates a new campaign or edit an existing campaign, then all the details about that can be seen in campaigns page .

**Input:** The input to this process are the details of a campaign created under a particular agency and advertiser by the user

**Output:** The details entered by the user get stored in our db and the campaign is ready to serve the purpose of UpSnap.

### **3.2.3 Edit existing campaign:**

This module allows a user to edit the already created campaign by just clicking on the edit button and entering the new details about that particular campaign . A user can only edit some of the allowed resource to those campaigns that he is allowed to.

**Input:** The input to this process is updated details of campaign that is created by the user.

**Output:** The input resource object is associated with the topic chosen by the user, also this resource is added to the reading items of all the users that have subscribed that topic.

### **3.2.4 Reports:**

This page shows all the data related to a running campaign

Profit , Loss can be calculated from this page. It also has a graphical Representation of the data.

**Input:** The input to this process is the credentials of user.

**Output:** You can view the report and change your campaign details according to it

### **3.2.5 Manage User Profile:**

This module allows a user to manage his profile details. A user can change and view his details at any time after he has authenticated successfully.

**Input:** The input to this process is a user object that contains the various changes that are made by the user in his profile details.

**Output:** The input user object is validated to check if all the details that are entered by the user are correct or not. If these details are correct, then the changes are saved in the database, otherwise an error message is shown and user is prompted to correct those details.

## **3.3 Activity :-**

**1) User Login Activity :** After clicking on the login a pop-up window will open user can choose the way by which he want to login.

**2) Registration Activity :**

User can register via filling his/her details. After that a mail with verification link is sent to their given email id after clicking on this link he/she is required to fill additional details and password after that he/she will be successfully registered.

### 3.4 Schema Design

IDM uses following Table Structures for storing the information in a database. Each table belongs to a particular entity or domain in the application. Like for a Campaign there is a user table which stores data of each and every campaign details same as for agency , advertiser.

Schema	List of relations Name	Type	Owner
public	address	table	postgres
public	advertiser	table	postgres
public	advertiser_pixel_tracker	table	postgres
public	advertiser_to_address	table	postgres
public	advertiser_to_contacts	table	postgres
public	agency	table	postgres
public	agency_addresses	table	postgres
public	agency_to_contact	table	postgres
public	authorities	table	postgres
public	caapi_user	table	postgres
public	caller_analytics	table	postgres
public	caller_daily_mail_report	table	postgres
public	campaign	table	postgres
public	campaign_analytics	table	postgres
public	campaign_daily_report	table	postgres
public	campaign_daily_report2	table	postgres
public	campaign_tracker	table	postgres
public	campaign_zip_code_group_mapping	table	postgres
public	contact	table	postgres
public	contact_to_person	table	postgres
public	cron_tracker	table	postgres
public	entity_theme	table	postgres
public	hibernate_sequences	table	postgres
public	prod_idm_psql_1805021132	table	postgres
public	user_analytics	table	postgres
public	user_demographics	table	postgres
public	user_profile	table	postgres
public	user_profile_advertisers	table	postgres
public	user_profile_campaigns	table	postgres
public	zip_code_group	table	postgres

(30 rows)

Figure.3.1 : Tables UpSnap



The description of the individual tables is as follows:

**Address:**

This table stores address about all the users that are registered in the Application.

Table "public.address"					
Column	Type	Modifiers	Storage	Stats target	Descript
id	bigint	not null	plain		
version	integer	not null	plain		
address1	character varying(255)	not null	extended		
address2	character varying(255)		extended		
city	character varying(255)	not null	extended		
formatted	character varying(255)		extended		
geo_hash	bigint		plain		
lat	double precision		plain		
lng	double precision		plain		
state	character varying(255)	not null	extended		
zip	character varying(255)	not null	extended		

Figure.3.2 : Address Table

**Advertiser:**

This table stores information about the various advertisers details that will create agency and its campaigns under it

Table "public.address"					
Column	Type	Modifiers	Storage	Stats target	Descript
id	bigint	not null	plain		
version	integer	not null	plain		
address1	character varying(255)	not null	extended		
address2	character varying(255)		extended		
city	character varying(255)	not null	extended		
formatted	character varying(255)		extended		
geo_hash	bigint		plain		
lat	double precision		plain		
lng	double precision		plain		
state	character varying(255)	not null	extended		
zip	character varying(255)	not null	extended		

Figure.3.3 : Advertiser Table

## Agency Table:

This table stores information about the various agencies created under a particular advertiser

Column	Type	Table "public.agency" Modifiers	Storage	Stats target	D
agency_id	bigint	not null default nextval('agency_agency_id_seq'::regclass)	plain		
version	integer	not null	plain		
active	boolean		plain		
billing_type	character varying(255)		extended		
contact_name	character varying(255)		extended		
cost_per_piece	double precision		plain		
created_date	timestamp without time zone	not null	plain		
description	character varying(255)		extended		
email_id	character varying(255)		extended		
name	character varying(256)	not null	extended		
phone_number	character varying(255)		extended		
large_size_card_cost	double precision		plain		
medium_size_card_cost	double precision		plain		
IntelliJ IDEA Ultimate Edition	double precision		plain		
agency_logo	character varying(255)		extended		
agency_subdomain	character varying(255)		extended		
primary_colour	character varying(255)		extended		
secondary_colour	character varying(255)		extended		
created_by	bigint		plain		
date_created	timestamp without time zone		plain		
last_updated	timestamp without time zone		plain		
updated_by	bigint		plain		

Figure.3.4 : Agency Table

## Caller Analytics Table:

This table stores information about the user which are using the facility through phone calls.

Column	Type	Table "public.caller_analytics" Modifiers	Storage	Stats target	Description
id	bigint	not null	plain		
call_id	character varying(255)		extended		
caller_phone	character varying(255)		extended		
campaign_id	bigint		plain		
campaign_status	integer		plain		
created_date	timestamp without time zone	not null	plain		
first_call_time	timestamp without time zone		plain		
is_mailable	boolean		plain		
total_calls	bigint		plain		
updated_date	timestamp without time zone		plain		
advertiser_id	bigint	not null	plain		
zip_code	double precision		plain		

Figure.3.5 : Caller Analytics Table

### Campaign\_Daily\_Report Table:

This table stores information about the various campaign visiting daily date wise in the System.

Column	Type	Modifiers	Storage	Stats target	Description
id	bigint	not null default nextval('campaign_daily_report_id_seq'::regclass)	plain		
created_by	bigint		plain		
date_created	timestamp without time zone		plain		
last_updated	timestamp without time zone		plain		
updated_by	bigint		plain		
version	bigint	not null	plain		
count	integer		plain		
date	timestamp without time zone		plain		
campaign_id	bigint		plain		
sent_count	integer		plain		

Figure.3.6 : Campaign Daily Report Table

## Campaign Table:

This table stores information about the various campaigns that were created in campaigns page under a agency and advertiser in the System.

Column	Type	Table "public.campaign" Modifiers	Storage	Stats target	Descriptio
id	bigint	not null default nextval('campaign_id_seq'::regclass)	plain		
created_by	bigint		plain		
date_created	timestamp without time zone		plain		
last_updated	timestamp without time zone		plain		
updated_by	bigint		plain		
version	bigint	not null	plain		
active	boolean		plain		
back_image	character varying(255)		extended		
back_image_name	character varying(255)		extended		
budget	double precision		plain		
campaign_url	character varying(255)		extended		
end_date	timestamp without time zone		plain		
front_image	character varying(255)		extended		
front_image_name	character varying(255)		extended		
name	character varying(255)	not null	extended		
post_card_size	integer		plain		
price_per_piece	double precision		plain		
product	integer		plain		
run_status	integer	not null	plain		
start_date	timestamp without time zone	not null	plain		
advertiser_id	bigint	not null	plain		
mail_count	integer		plain		
max_mailers_day	integer		plain		
max_mailers_month	integer		plain		
billing_budget	double precision		plain		
remain_budget	double precision		plain		
exclude_url_arr	text[]		extended		
include_url_arr	text[]		extended		
allocated_number	character varying(255)		extended		
website	character varying(255)		extended		
billing_increment	integer		plain		
reload_amount	double precision		plain		
last_paused_at	timestamp without time zone		plain		
ended_on	timestamp without time zone		plain		

Figure.3.7 : Campaign Table

## CHAPTER-4: IMPLEMENTATION AND CODING

The purpose of this chapter is to describe the languages, packages and backend used in the project. What should be tested in testing phase and what are the validations applied on the project.

### 4.1 System Development using MVC architecture

The development of the application was done by following the MVC architecture in Spring. MVC architecture divides an application into three parts which can be configured independently of each other, these parts are:

#### **Model:**

It deals with the entities that are present in the database. It handles how the data is stored and retrieved from the database.

#### **View:**

View part of an application deals with how data is displayed in a client. Typically a client is a web browser for web applications. In this case, View layer deals with how the data will be rendered as HTML.

#### **Controller:**

Controller layer deals with what part of Model and View will be used to handle a client request. It also supervises the flow of a request from any client and how that request is handled by the server and what response is generated and transmitted back to the client.

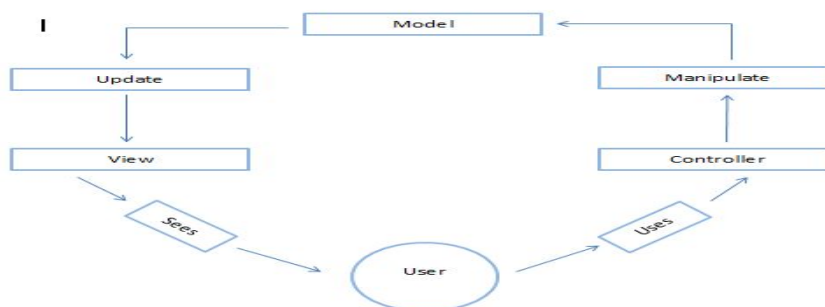


Figure.4.1 : MVC Structure

## 4.2 Spring Directory Structure

The application is developed in Spring framework. Following is the directory structure of a Spring application:

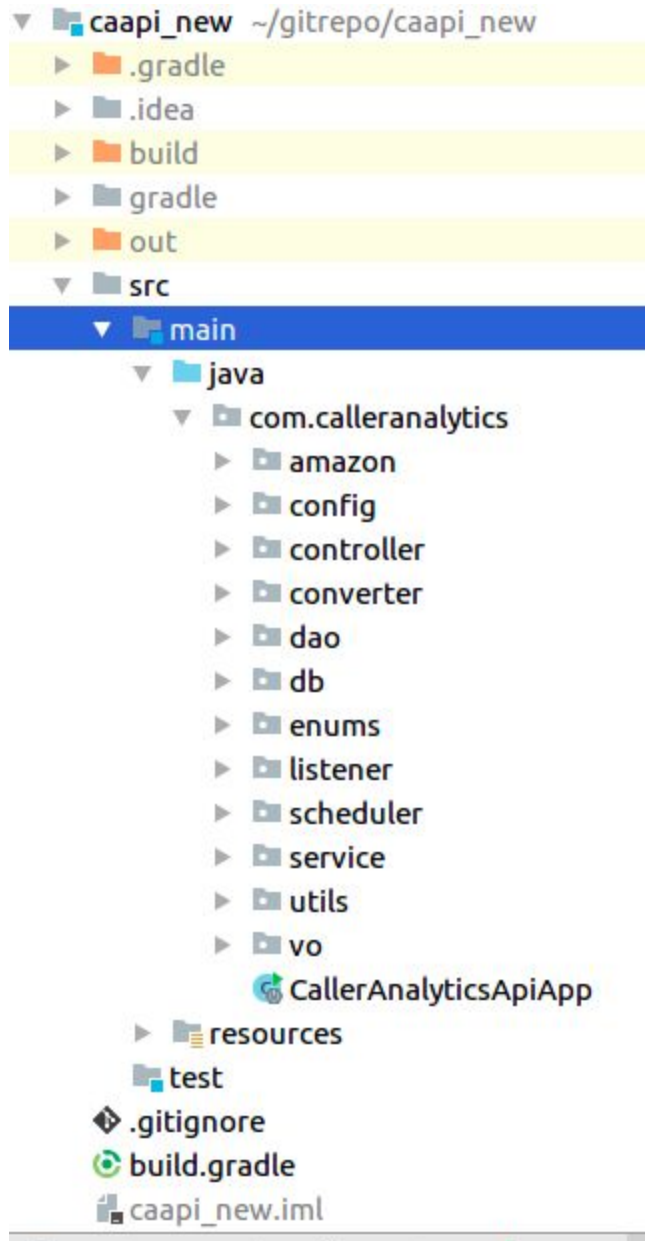


Figure. 4.2 : Spring Directory Structure

## 4.3 High Level Design

Various Classes for the MVC architecture in grails are as follows:

**4.3.1 Model:** In Spring, Domain Classes and Services collectively play role of Model.

### Domain Classes:

The data that is needed to be persisted on secondary memory is encapsulated into Domain classes. Typical example of Domain classes are User, Campaign, Agency, Advertiser, etc. Every domain class is mapped to a table in database with their fields as columns.

Following are the domain classes that were created in application.

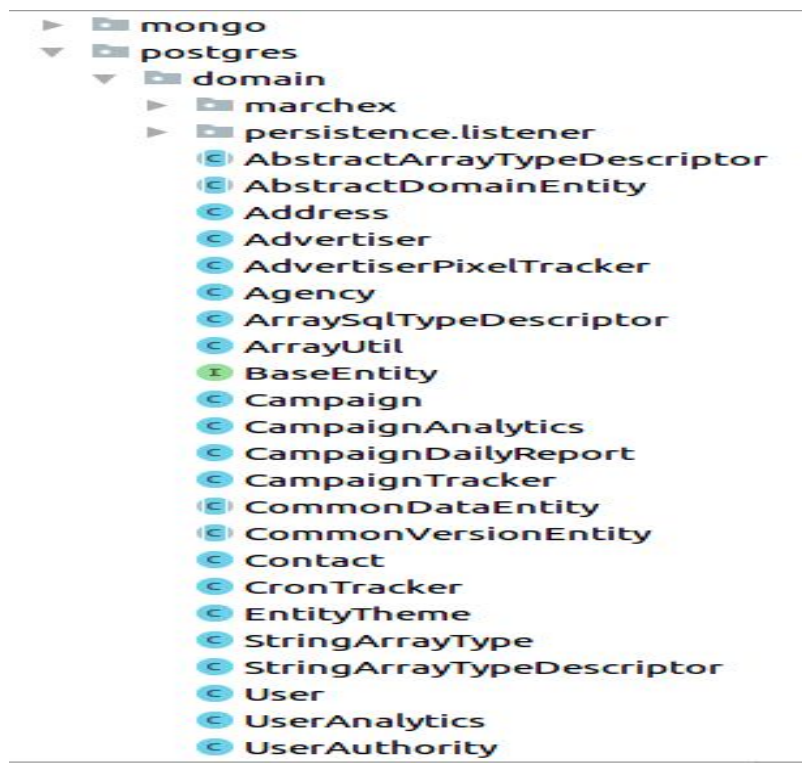


Figure. 4.3 : Domain Classes

## Services

Spring framework provides out of the box transactional support. Services are nothing more than Classes. The only thing that is special about services is that we get transactional behavior. Programmers don't need to worry about the code for this, everything is managed by the framework. Programmers need to work only on business logic of the application.

Various service classes created application are:

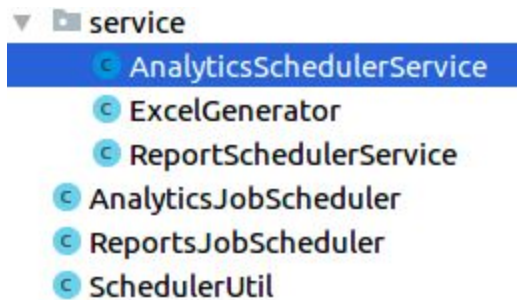


Figure. 4.4 : Services

### 4.3.2 Views:

In MVC applications, views are the components that interact with user. In typical java applications, views are made in html, xhtml, jsp, etc. These are similar to Java Server Pages and provide full support for client side scripting and styling.

Various views created in application are:



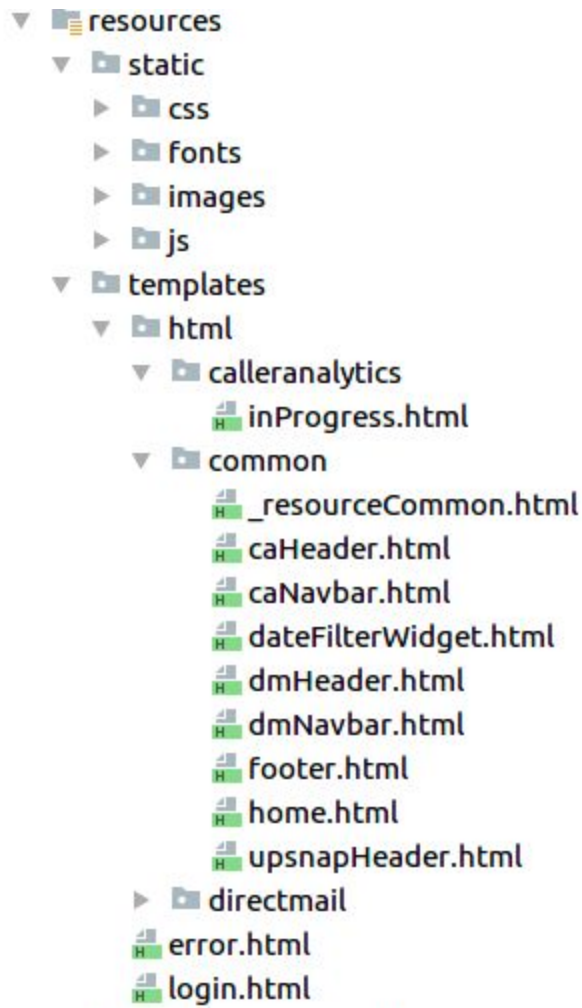


Figure. 4.5 : Views

### 4.3.3 Controllers:

As their name suggest, their main task is to control. With the help of services and domain classes, controllers process incoming requests from clients and send appropriate responses.

Various Controllers created in our application are:



Figure. 4.6 : Controllers

## 4.4 Methodologies and Tools

### 4.4.1 Technology Stack Used

#### 1) Apache Tomcat:-

Apache Tomcat (or Jakarta Tomcat or simply Tomcat) is an open source servlet container developed by the Apache Software Foundation (ASF). Tomcat implements the Java Servlet and the JavaServer Pages (JSP) specifications from Sun Microsystems, and provides a "pure Java" HTTP web server environment for Java code to run.

2) **Spring**:-Central to the Spring Framework is its [inversion of control](#) (IoC) container, which provides a consistent means of configuring and managing Java objects using [reflection](#). The container is responsible for managing [object lifecycles](#) of specific objects: creating these objects, calling their initialization methods, and configuring these objects by wiring them together.

Objects created by the container are also called managed objects or [beans](#). The container can be configured by loading [XML](#) (Extensible Markup Language) files or detecting specific [Java](#)

**annotations** on configuration classes. These data sources contain the bean definitions that provide the information required to create the beans.

Objects can be obtained by means of either dependency lookup or dependency injection.<sup>[12]</sup> Dependency lookup is a pattern where a caller asks the container object for an object with a specific name or of a specific type. Dependency injection is a pattern where the container passes objects by name to other objects, via either **constructors, properties, or factory methods**.

In many cases one need not use the container when using other parts of the Spring Framework, although using it will likely make an application easier to configure and customize. The Spring container provides a consistent mechanism to configure applications and integrates with almost all Java environments, from small-scale applications to large enterprise applications.

The container can be turned into a partially compliant **EJB (Enterprise JavaBeans) 3.0** container by means of the Pitchfork project. Some<sup>[who?]</sup> criticize the Spring Framework for not complying with standards.<sup>[13]</sup> However, SpringSource doesn't see EJB 3 compliance as a major goal, and claims that the Spring Framework and the container allow for more powerful programming models.<sup>[14]</sup> The programmer does not directly create an object, but describe how they should be created, by defining it in the Spring configuration file. Similarly services and components are not called directly; instead a Spring configuration file defines which services and components must be called. This IoC is intended to increase the ease of maintenance and testing.

Spring is powered by the most popular open source technologies in their respective categories:-

*Hibernate*: The de facto standard for ORM in the Java world.

*Quartz*: An enterprise ready, job scheduling framework allowing flexibility and durable execution of scheduled tasks.

*Site Mesh*: A robust and stable layout rendering Framework.

ORM simply serves as a way to map objects from the object oriented world onto tables in a relational database. ORM provides an additional abstraction above SQL, allowing developers to think about their domain model instead of getting wrapped up in reams of SQL. IoC, also known as dependency injection, is a way of —wiring” together objects so that their dependencies are

available at run time. As an example, an object that performs persistence may require access to a data source. IoC provides a way to take the responsibility of obtaining a reference to the data source off the developer.

### **3) jQuery :-**

jQuery is a lightweight Javascript library that emphasizes interaction between Javascript and HTML. It was released January 2006 at BarCamp NYC by John Resig . The jQuery library provides a generalpurpose abstraction layer for common web scripting, and is therefore useful in almost every scripting situation.

#### **4.4.2 Software Tools Used:**

##### **1) Version Control System: Git**

In computer software engineering, revision control is any practice that tracks and provides control over changes to source code. Software developers sometimes use revision control software to maintain documentation and configuration files as well as source code.

As teams design, develop and deploy software, it is common for multiple versions of the same software to be deployed in different sites and for the software's developers to be working simultaneously on updates. Bugs or features of the software are often only present in certain versions (because of the fixing of some problems and the introduction of others as the program develops). Therefore, for the purposes of locating and fixing bugs, it is vitally important to be able to retrieve and run different versions of the software to determine in which version(s) the problem occurs.

It may also be necessary to develop two versions of the software concurrently (for instance, where one version has bugs fixed, but no new features (branch), while the other version is where new features are worked on (trunk).

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. Git's design is a synthesis of Torvalds's experience with Linux in maintaining a large distributed development project, along

with his intimate knowledge of file system performance gained from the same project and the urgent need to produce a working system in short order.

## 2) Project Development Tracking System

In order to track development and progress of system, there are many tracking systems available. Some of them are: Jira, Asana, Trac. We use Trac. In Trac work is assigned to team members in the form of tickets. Client creates tickets for every task that he wants to be done by development team. These tickets are then assigned to developers. Developers work on the tickets and when their task gets completed, client tests the task and closes the ticket.

Various phases in life-cycle of a ticket are:

Open: When a ticket is just created it is called as open ticket.

Assigned: In this phase the ticket is assigned to a team member and he is working on it.

Ready for testing: It is the stage when team members have completed their task and deployed on QA server. At this stage, ticket is ready for acceptance testing. Client tests the ticket on QA server and if tickets do not pass the test it is again rendered as open ticket.

Ready for release: If ready for testing tickets pass all the acceptance tests, it is rendered as ready for next release. During the release process, the code is actually deployed on production environment.

Closed: After the ticket is deployed on production environment, a ticket is rendered as closed.

Also, tickets can have priorities assigned to them. Low, Normal, High, Critical, Very critical. Team members are supposed to work on high priority tasks first.

If team member or client want to communicate something about any particular ticket, they can add their comments and even attach files.

Also, Track shows a number of reports to track development process. For eg to show:

- all the assigned tickets.
- all the closed tickets.
- all the tickets done by a particular team member.

## **CHAPTER-5: TESTING**

Testing is the process of exercising the software item to detect the differences between its behavior and the desired behavior as stipulated by the requirements specifications– ‘\_what is’ and ‘\_what should be’. Testing is a destructive process. To achieve a very high standard in quality of delivery, a comprehensive and planned testing will be carried out during project execution.

Various types of Testing would be carried out by IntelliGrape System (P)Ltd. in the SDLC phases to test various aspects of the software.

### **5.1 Unit Testing:-**

#### **Definition of a unit**

- o Screens with its associated services, which help to display, add, modify, delete, authorize or list data.
- o Any function, which encapsulates business logic for a batch job or a service.
- o Batch jobs separated as a unit
- o Library functions being provided by any module to be considered as a unit.

#### **Aim of Unit Testing**

- o To test a unit of code after coding of that unit is completed.
- o To utilize a finite number of test cases to detect the maximum number of errors using minimum resources.
- o To achieve maximum test coverage.

## **Unit Testing Constituents**

- To check if the unit functionally and technically achieves its purpose in completeness and correctness. For instance, Code Coverage, Memory leakage, As per System specs and program specs.

## **Prerequisite to Unit Testing**

- Unit test Plan and Unit Test specifications reviewed by QA are available.
- Reviewed code with all the identified defects fixed.
- The data setup is in place for the unit testing
- Stubs used by the unit are ready.

## **Unit Testing Process**

- Follow the Unit Test Specifications (UTS) to test the items and log the results in the Unit Test Results document.
- For defects being encountered during testing, the same would also be logged in IPMS (Integrated Project Management system), the inhouse project management software.
- The tester would fix the defects after one round of testing is over. The fixes would be made on the unit that has been checked out of the controlled environment if it is an old item.
- To close the defects in IPMS the tester would then do a regression testing in the development environment.
- After the tester has successfully executed all the test conditions and achieved the desired code coverage, the unit would be introduced into the QA environment.
- The QA team would then carry out a round of testing to verify the results and match them with the desired functionality. If discrepancies are found then defects are logged into IPMS and fix cycle is repeated.
- Once the QA team approves the unit testing, then the item is ready to be moved to system testing environment.



## **5.2 Module Testing :-**

Also known as unit or component testing phase, module testing is concerned with the testing of the smallest piece of software for which a separate specification exists. Both IEEE 1008 and BS 79252 detail the test process required at this level, while BS 79252 also defines the test case design techniques which can be applied. When designing and implementing large scale software one of the most important issues in order to reduce the cost and improve the quality is testing. In this master thesis a thorough background to testing with the focus on low-level testing is given as an introduction. Ideas and theory for different kinds of testing as well as available tools are described. The thesis focus on low-level testing and this is important because it saves both time and costs in the later phases of software development. Today's development teams become more and more aware of the importance of low-level testing. Unfortunately low-level testing is often inefficiently deployed as it is poorly structured and there are very few tools. To facilitate an raise the quality of the low-level testing this thesis presents a process for module testing as well as describes the development of an automated testing tool. The process should ensure that all module tests follow the same structure and that all of the modules get the same type of low-level testing. The automated testing tool is a prototype that makes the procedure automated and saves time. The thesis also shows that the developed process and tool greatly enhanced and improved the software quality in the projects using it.

## **5.3 Integration Testing:-**

### **Definition of an Integrated Unit**

- o A Batch job
- o All the screens of a particular module.
- o All screens which provide the facility to access screens of another module.

### **Aim of Integration Testing**

- o To prove that the various programs making up the system are compatible, fit together and the interfaces between the programs are correct

### **Integration Testing Constituents**

- o To check if the integrated unit technically and functionally achieves its purpose.

### **Prerequisite to Integration Testing**

- o Integration test plan and specifications reviewed by QA are available
- o Unit tested and verified code
- o Compile scripts for the units under the Integrated Unit

### **Integration Testing Process**

- o The data setup is in done for the integration testing
- o Follow the Integration test specifications to test the items and log the results in the Integration test results document.
- o For defects being encountered during testing, the same would also be logged in IPMS, the in house project management system. The tester would fix the defects and do a retes by executing all the test conditions and once convinced that the defect is fixed, the defects are closed in IPMS. The fixes would be made on the unit that has been checked out of the controlled environment if it is an old item.
- o The unit would be promoted to the QA environment.

- o The QA team would then carry out testing against the integration test specifications and verify the results. If discrepancies are found then defects are logged into IPMS and fix cycle is repeated. QA team will also do a regression testing to ensure that fix has not created any unexpected result
- o Once the QA team approves the integration testing, then the item is ready to be moved to system testing environment. The same can also be released for interim delivery, whenever required

### **Integration Testing Deliverables**

The deliverables from the Integration testing will be:

- o Integration test logs
- o Tested code

## **5.4 Functional Testing**

The objective of this test is to ensure that each element of the application meets the functional requirements as outlined in system specifications. Before executing the system test cases in full, a limited functional testing will be performed with a subset of system test case where the system will be run on two (or may be more is to be decided) business days and covering end to end two (or may be more is to be decided) event types. This is done to verify if all the components of the system is installed properly and to do a basic functionality testing. This will conclude high-level testing. It will be followed by detailed level tests, which will aim to test the individual processes and data flows.

### **Interface Testing**

The various programs that facilitate the exchange of data between the NCS and other interfacing systems will be tested through the Interface sub testing. The process of Interface testing will be interlinked to the system testing activity,

by incorporating the messaging requirements within the system test conditions. Non-functional Testing: Besides testing to check whether the application meets all the functional requirements, following Non-functional Testing would be carried out in this phase:

### **Installation Testing**

- o To check ease of installation & that the Installation can be carried out as per the installation Manual
- o To check whether data migration is done successfully
- o To check whether Installation can be done on multiple platforms (for instance on AIX and Mainframe)

### **Security Testing**

- o To check whether the system can be hacked
  - o To test how the application recovers in cases of failures of the supporting architecture
- ### **User Interface Testing**
- o To check the ease of use in cases of large volumes of data
  - o To check the ease of usage of the User Interface under Normal loads.

### **Consistency Testing**

- o To check whether the system gives the same results on repeating Test runs.
- o To check whether the application is consistent in terms of data, and look and feel.

## CHAPTER 6. CONCLUSION

The project work assigned to us has been completed to our satisfaction and we have added significant knowledge to ourselves. This System has some limitations which can be removed by making enhancements to the System.

### 6.1 Value Addition to Knowledge

**Domain Experience:** IntelliGrape is a pioneer in all phases of a web application development. As each trainee is assigned many responsibilities, it has made us learn a lot.

**Exposure to entirely different technology:** Since we worked with rails, which was a new technology for us, we have really enriched our knowledge significantly.

**Software development process, methodologies and standards:** This project allowed us to work as an agile team; therefore we could understand how it actually manages the flow of development by staying in sync with the requirements of the client.

### 6.2 Value Addition to the project?

**Implementation of Business Logic Layer (BLL):**We have implemented the classes for BLL, which provides the interface between UI Layer and Data Access Layer. This layer contains all the business logic, i.e. programs that implement the functionality of the system.

**Implementation of User Interface Layer (UI Layer):**We have also designed the UI layer which contains the web pages designed in GSP and Java Script. This also contains the code behind pages that implement the programming logic.

### 6.3 Future Scope for Modification

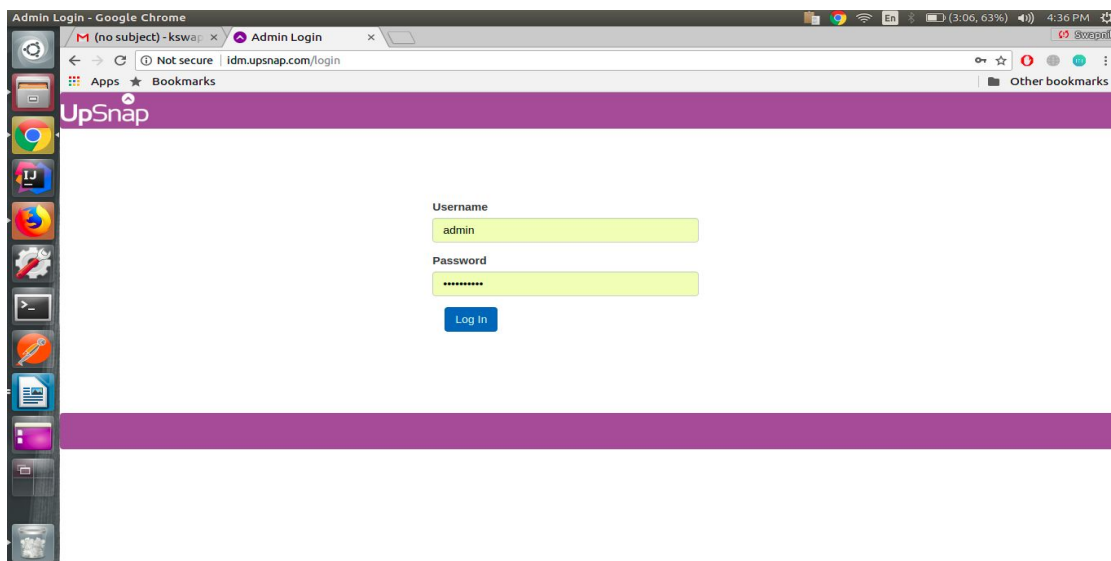
IDM has very vast scope of future enhancement. In future this website can be used as question/answer portal which provide quality learning material and also let user ask questions and allow them to comment.

It may have following future enhancements :

- In future it can allow users to comment upon the topics.
- In future there can be multilingual support for users by which users who are not comfortable in English don't find the problems to understand the topics.
- Publishing Report can become fast by giving rights of review to some of contributors who are well known in the appropriate fields.
- More registration option can be provided like Twitter/Google+ etc.

### Interface Designs (Screen Shots)

- **Login Page of the Site :-**



- **User Dashboard**

The screenshot shows the 'User Detail' page in the UpSnap application. The page has a purple header with the UpSnap logo and navigation tabs for Users, Agencies, Advertisers, Campaigns, and Reports. A 'New User' button is located in the top right corner. Below the header, there is a 'User Detail' section with a search bar and a table listing users. The table has columns for User ID, User Name, Mail Id, Role, Status, and Edit. There are 8 users listed, all with an 'Active' status.

User ID	User Name	Mail Id	Role	Status	Edit
1	Lance	admin	SUPER_ADMIN	Active	Edit
2	Jivon	jmackey@upsnap.com	ADMIN	Active	Edit
3	Ron	rfoster@upsnap.com	ADMIN	Active	Edit
4	Lance	lbrown@ufpb.net	AGENCY_ADMIN	Active	Edit
5	Sarah	sfarquhar@upsnap.com	SALES	Active	Edit
6	Sample	sample@upsnap.com	USER	Active	Edit
7	Tina	tkarel@upsnap.com	SALES	Active	Edit
8	Beth	bmanning@ufpb.net	ADMIN	Active	Edit

- **Agency Dashboard**

The screenshot shows the 'Agency Detail' page in the UpSnap application. The page has a purple header with the UpSnap logo and navigation tabs for Users, Agencies, Advertisers, Campaigns, and Reports. A 'New Agency' button is located in the top right corner. Below the header, there is an 'Agency Detail' section with a search bar and a table listing agencies. The table has columns for Agency, Agency ID, Eligible Visitors, Total Mailers Sent, Spend, Cost, and Edit. There are 8 agencies listed with various metrics.

Agency	Agency ID	Eligible Visitors	Total Mailers Sent	Spend	Cost	Edit
25th Hour Communications (UpSnap)	35	5462	489	\$592.20	\$333.27	Edit
Cisneros Media Partners (UpSnap)	34	159	157	\$235.50	\$72.22	Edit
Forecasting Only	3	134				Edit
GoLocal247.com	4	2				Edit
Hibu	38	0				Edit
Mandile Marketing (UpSnap)	5	7496	5005	\$7507.50	\$2302.30	Edit
UpSnap	1	28739	4013	\$6429.06	\$2290.77	Edit

## Advertiser Dashboard

Advertiser Detail

Show 25 entries Search:

Advertiser	Advertiser ID	Eligible Visitors	Total Mailers Sent	Budget	Total Spent	Edit	Status
1-800-2 Sell Homes - 082153u	202	901	662	\$1024.00	\$993.00	Edit	03-14-2018
25th Hour - AACC Convention	341	23	18	\$534.00	\$27.00	Edit	05-04-2018
25th Hour Lake Region State College	352	0				Edit	04-25-2018
806exteriors.com - 076151u	248	34	15	\$1000.00	\$30.00	Edit	05-03-2018
A G Auto Care - 063992u	55	0	7	\$500.00	\$14.00	Edit	01-30-2018
A to Z Contracting Co - 055079u	297	28	15	\$1250.00	\$37.50	Edit	04-13-2018

## Campaign Dashboard

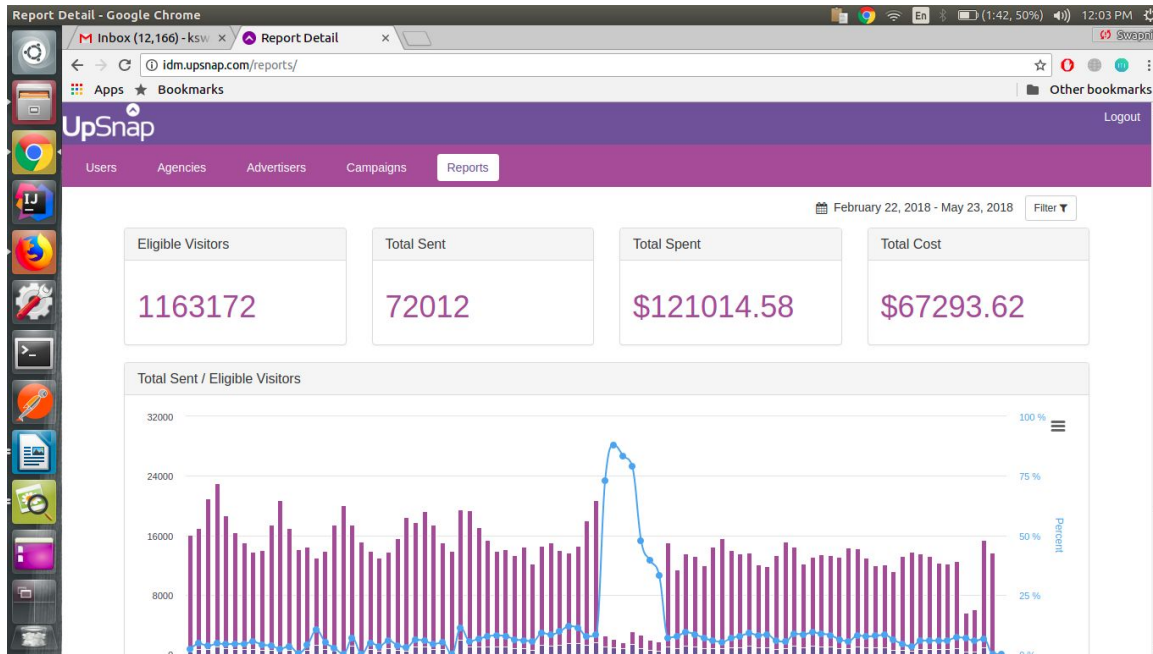
Campaign Detail

Show 25 entries Search:

Advertiser	Campaign	Campaign Id	Campaign Product	Status	Start Date	End Date	Eligible Visitors	Total Mailers Sent	Budget	Total Spent	Edit	Resume
1-800-2 Sell Homes - 082153u	1-800-2 Sell Homes - 360557u	178	WEB_TARGETING	LIVE	03/15/2018	03/15/2019	866	662	\$1024.00	\$993.00	Edit	
25th Hour - AACC Convention	25th Hour - AACC Convention	306	WEB_TARGETING	ENDED	04/28/2018	05/01/2018	20	18	\$534.00	\$27.00	Edit	
806exteriors.com - 076151u	806exteriors.com - 360851u	242	WEB_TARGETING	LIVE	04/02/2018	04/02/2019	24	15	\$1000.00	\$30.00	Edit	
A G Auto Care - 063992u	A G Auto Care - 359511u	30	CALL_RETARGETING	LIVE	02/01/2018	02/01/2019	0	7	\$500.00	\$14.00	Edit	
A to Z Contracting Co - 055079u	A to Z Contracting Co - 361250u	285	WEB_TARGETING	LIVE	04/13/2018	04/13/2019	28	15	\$1250.00	\$37.50	Edit	



- **Report Dashboard**



## References/Bibliography

### 1: Grails Plug-ins Used

- |                             |          |   |
|-----------------------------|----------|---|
| 1) <i>Quartz</i>            | Purpose: | For creating asynchronous Jobs  |
|                             | Version: | quartz-1.0-RC5  |
| 2) <i>Asynchronous Mail</i> | Purpose: | For sending mails asynchronously  |
|                             | Version: | asynchronous-mail-1.0-RC3   |
| 3) <i>jQuery</i>            | Purpose: | For using jQuery  |
|                             | Version: | jquery-1.8.0  |
| 4) <i>Console</i>           | Purpose: | It provides easy access to database, configuration files which is very much useful while debugging our application. |
|                             | Version: | console-1.2   |

### 2: User Interface Plug-ins used

- 1) Twitter Bootstrap : It provides Sleek, intuitive, and powerful front-end framework for faster and easier web development.
- 2) jQuery UI : jQuery UI is a set of user interface interactions, effects, widgets, and themes built on top of the jQuery JavaScript Library. Whether you're building highly interactive web applications or you just need to add a date picker to a form control.

### **3: Books**

1. Spring In Action A Practical Guide to Feature Driven Development Stephen R. Palmer Wesley, 1999.
2. Extreme Programming in Practice , James Newkirk and Robert C. Martin, Addison Wesley, 2001
3. Beginning Spring Christopher M. Judd, Joseph Faisal Nusairat, and James Shingler
4. Learning JQuery Better Interaction Design and Web Development with Simple JavaScript Techniques by Jonathan Chaffer Karl Swedberg

### **4: URLs:**

1. Spring Website : [www.spring.io](http://www.spring.io)
2. Redis Website : [www.redis.io](http://www.redis.io)
3. jQuery Website : <http://jqueryui.com/>
4. Twitter Bootstrap from : <http://twitter.github.io/bootstrap/>
5. Mysql tutorials : [www.mysql.com](http://www.mysql.com).
6. <http://en.wikipedia.org/>