

Abstract

”Eventati” is a Social Event Directory not only a website but also a mobile application to manage and deal with events via internet, provides with most of social functionalities by 25Jan revolutionists to all eventatiens.Because of problems that face most of people when they miss some of his important events in all life fields .So ”Eventati” aimed to collect all events in one site, get localization of each event through map, and to allow communications between all friends. ”Eventati” also allows users to invite their friends and follows others and also provides the ability to create their own events, invite their friends to attend events, share events among their friends what make the events spread widely. ”Eventati” automates a lot of repeating tasks and collecting the data together in one place where you can access any time.

Mobile Application describes a feasible way of integrating mobile networks with the Internet, in order to deal with events that allow mobile devices to receive common web transactions and allows Internet peers to be notified of mobile generated events. Mobile application of ”Eventati” facilitates accessing for mobile users to access the events through their devices, while Mobile applications are hugely popular for customer service. We thought in mobile application as a result of the great revolution in mobile application development and we have finished developing our application. ”Eventati” web Team has finished developing user management system and event management system.

”Eventati” consists of three major teams. Every team uses open Source technologies. Firstly, Web Team uses three major tools, Django framework, Pinax with Django and python as a programming language. Secondly, Mobile Team uses Appcelerator Titanium with Java Script to create a version of the application for each targeted device. Finally, Design team uses advanced Graphic editing tool (Photoshop). In the future we will add some functions to our project such as Mozilla Firefox add-ons, aversion for tablets, procedures like how to reach to specific Event, enabling transportation and on-line payment for events to allow users to pay to get event’s tickets.

Acknowledgements

In the name of Allah the Most Gracious, the Most Merciful. We would like to express our eternal appreciations towards our beloved parents and family who have always been there for us, for all unconditional encouragement and patience. Thank you for being ever so understanding and supportive. We must give our high, respectful gratitude to our supervisor "Dr. Haitham El-Ghareeb" for the valuable guidance and advice. He inspired us greatly to work in this project. His willingness to motivate us contributed tremendously to our project. We also would like to thank him for showing us some example that related to the topic of our project.

We owe a great many thanks to many people who helped us and supported us during "Eventati" Project. Firstly, Endless thanks especially from the design team to Eng. Shaimaa Adel for the guidance and help throughout the project. We have learned a lot from her and she gave us precious advices in order to improve ourselves and become better designers. Secondly, The Mobile team and Web team tends thanking to Eng. Elsayed Gamal for his valuable time and effort and for supporting the whole group to work using open source technologies. Thirdly, Great thanks to Harvest and ILC company for being one of the sponsors to the project by providing labs and convenient place.

Contents

1	Introduction	1
1.1	Need for the Project	1
1.2	Aim of the Project	1
1.3	Integration between Mobile and Web	2
1.4	Thesis Structure	2
2	Methodologies	5
2.1	Agile Methodologies	5
2.2	SCRUM	5
2.2.1	Meetings	7
2.2.2	Stories:	11
2.2.3	Testing:	22
3	Technologies	23
3.1	Web Technology	23
3.1.1	Python	23
3.1.2	Django	24
3.1.3	The MVC Design Pattern	25
3.1.4	Pinax	29
3.2	Mobile Technology	31
3.2.1	Appcelerator	31
4	Web Components	33
4.1	Introduction	33
4.2	System Requirements	33
4.2.1	Publish Event	33
4.2.2	Find My friends	34
4.2.3	Update TimeLine	39
4.3	website back-end design	46
4.4	Interface Design	51
4.4.1	Landing page	51

4.4.2	Home page	51
4.4.3	Create Events	53
4.4.4	Profile Page	55
4.4.5	Setting Page	55
4.5	Implementation of testing	55
4.5.1	Testing	55
4.5.2	Why testing ?	59
4.5.3	When testing?	59
4.5.4	Writing testing	59
4.5.5	Handelling Testing Of Eventati	60
5	Mobile Components	61
5.1	Introduction:	61
5.2	User Requirements:	62
5.2.1	Update TimeLine	62
5.3	User Interaction with Mobile application:	64
5.4	Code of Mobile Application	64
6	Design Components	71
6.1	Introduction	71
6.2	Logo Design	71
6.3	Web pages layouts	73
6.3.1	Landing page	73
6.3.2	Home page	75
6.3.3	Create event page	75
6.3.4	Profile page	75
6.3.5	Event Details page	75
6.3.6	Search page	75
6.4	Design in Mobile Application	81
6.5	Mobile Icons	81
6.5.1	Mobile layout	81
7	Integration Between Mobile and web	85
7.1	Introduction	85
7.2	Integration Methods	85
7.3	XmlRPC	86
7.3.1	Overview	86
7.3.2	request Example	86
7.3.3	Header requirements	86
7.3.4	Payload format	87
7.3.5	Response format	87

7.3.6	Response Example	89
7.4	Code Examples:	89
7.4.1	Example from Xml File:	89
7.4.2	Example From UI Application Files	89
8	Outcomes	93
8.1	Web Outcome	93
8.2	Mobile Outcome	95
9	Conclusion and future Work	109

List of Figures

1.1	Statistic of Social Network in 2011	2
1.2	Social Networks	2
2.1	Waterfall and Scrum	6
2.2	SCRUM Diagram	8
2.3	Scrum Team Members	9
2.4	Product Backlog Of "Eventati"	9
2.5	Meeting 6	12
2.6	Meeting 7	13
2.7	Meeting 8	14
2.8	Meeting 9	15
2.9	Meeting 1 Only Team	16
2.10	Meeting 2 only Team	17
2.11	Meeting 3 only Team	18
2.12	Meeting 4 only Team	19
2.13	Meeting 10	20
2.14	Meeting 11	21
3.1	Python Logo	23
3.2	Django Logo	23
3.3	Django Request and Response	25
3.4	Django Application	26
3.5	Django Application Structure	26
3.6	MVC Design Pattern	27
3.7	models.py (the database tables)	27
3.8	views.py (the business logic)	28
3.9	urls.py (the URL configuration)	28
3.10	latest-events.html (the template)	28
3.11	pinax	29
3.12	Appcelerator Logo	31
3.13	Titanium Logo	31
3.14	Appcelerator of Application Sources File	32

4.1	System Requirements	34
4.2	Context Diagram	35
4.3	Web-UseCase-PublishEvent	36
4.4	Web-Activity-PublishEvent	37
4.5	Web-Sequence-PublishEvent	38
4.6	Web-Use Case-Find My Friends	40
4.7	Web-Activity-Find My Friends	41
4.8	Web-Sequence-Find My Friends	42
4.9	Web-Use Case-Update TimeLine	43
4.10	Web-Activity-Update TimeLine	44
4.11	Web-Sequence-Update TimeLine	45
4.12	Class Diagram	47
4.13	Entity Relationship Diagram	48
4.14	Entity Relationship Diagram 2	49
4.15	Application Model	50
4.16	python manage.py syncdb	50
4.17	Landing Page	52
4.18	Example Code Of last 5 Events	53
4.19	Code Of Calling Function	53
4.20	The html code of page	54
4.21	The final output	54
4.22	Code 1 Of create Event	56
4.23	Code 2 Of create Event	56
4.24	The Result Out Put	57
4.25	Code Of user's Profile	57
4.26	user's profile Page	58
4.27	user's profile Page	58
4.28	Docstrings	60
5.1	Mobile-Use Case-Update TimeLine	63
5.2	Mobile-Activity-Update TimeLine	65
5.3	Code of Landing Page	66
5.4	Code Of the Slides Images	67
5.5	Code of Login page	68
5.6	Code of Agenda	69
5.7	Code Of Event details	70
6.1	AscII Code Colors	72
6.2	First Logo	72
6.3	Second Logo	72
6.4	Final Logo Of Eventati	72

6.5	Slogan Of Eventati	72
6.6	First Logo	73
6.7	Second Logo	73
6.8	Final landing Page	74
6.9	Eventati Home Page	76
6.10	Create events Page	77
6.11	Profile Page	78
6.12	Event Details page	79
6.13	Search page	80
6.14	Hand Drawing Mobile Icons	81
6.15	Final Mobile Icons	81
6.16	Mobile Layout	82
6.17	Landing Page	83
6.18	Slide 1	83
6.19	Slide 2	84
6.20	Slide 3	84
7.1	Web and Mobile consumption per day	86
7.2	"Integration Between Mobile and Web"	87
7.3	"Xml RPC"	88
7.4	"Request Example Code "	88
7.5	"Response Code Example "	89
7.6	"Code Example "	90
7.7	"UI Application File "	90
7.8	"UI Application File2 "	91
7.9	Event Details Code	92
8.1	Land Page	96
8.2	Land Page After pressing login Button	97
8.3	Land Page After pressing Registration Button	98
8.4	Home Page	99
8.5	Create Event	100
8.6	Profile Page	101
8.7	Event Details	102
8.8	Search	103
8.9	Landing Page Of Mobile	104
8.10	Login Page Of Mobile	104
8.11	Sign-Up Page Of Mobile	105
8.12	Home Page Of Mobile	105
8.13	Event's Details	106
8.14	Profile Page Of Mobile	106

8.15 "Slide 1 Of Mobile	107
8.16 Slide 2 Of Mobile	107
8.17 Slide 3 Of Mobile	108
8.18 Slide 4 Of Mobile	108

List of Tables

4.1	Description of Publish Event Process	35
4.2	Description of Find My Friends Process	39
4.3	Description of Update TimeLine	46
5.1	Description of Update TimeLine For Mobile	62

Chapter 1

Introduction

1.1 Need for the Project

Today and after the noted increasing in events in different fields of life ,it became difficult to us to handle these events manually and if we try that we will miss some of those events ,so thought about tool that handles our events and brings it where we are. This project tends to social Network for its important. Statistically By March 10-2011, there were more than 24 million FaceBook users in the Arab World [Arab Advisor Groups] , Figure 1.1 presented in page 2 shows Statistic of Social Network in USA,Jordan and Egypt. Social Network Sites like FaceBook, Twitter, Google+, LinkedIn , MySpace and REDDIT are in every where ,Figure 1.2 presented in page 2 shows logos of most famous social Networks[avtec].Today's youth are spending a great deal of time using these sites to access virtual life and also after revolution of 25 Jan Social Network Sites are widely increasing and also from our belief with importance of communication with others. The problem statement we noticed is when anyone wants to announce about event there is no dedicated place that s/he can invite people through it, he simply makes to every event websites or Web-page and to avoid missing events that I want to attend, So that we need "Eventati".

1.2 Aim of the Project

After the revolution of internet technology we found that created web and mobile application especially social website become important component in the life of every one as we with this technology can follow events in any field we have interested in it. After creating this project everyone can use this project to create his event and share it with his friends easily in ef-

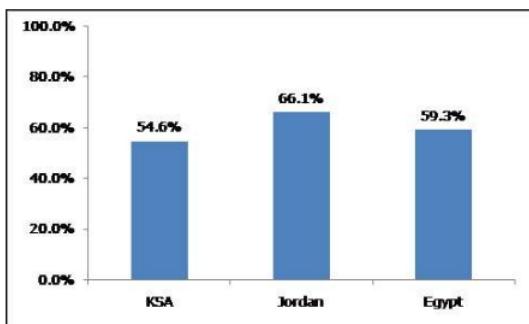


Figure 1.1: Statistic of Social Network in 2011



Figure 1.2: Social Networks

ficient way and also become a follower for any event that takes place around you.”Eventati” makes life easier to you, so it can enable you following your interested events where and when ever they are held.

1.3 Integration between Mobile and Web

To complete the purpose of our project (bringing events where we are),thinking in integration between our web application and mobile application as it easier for anyone to take mobile device with him/her in any place than taking personal computer and the using of mobile application very simple for any user,So integration is important. Mobile application must connect to web services (Data Base) which in the web application. By using standard protocols like SOAP protocol by reading WSDL by XML. Message.response representations from the according web service protocols (WSDL, SOAP, and UDDI) have to be translated into corresponding requests response data types of the agent system, and vice versa.To integrate between web and mobile application by using XML RPC. XML RPC: It’s a set of implementations that allow software running on disparate operating systems, running in different environments to make procedure calls over the Internet. It’s remote procedure calling using HTTP as the transport and XML as the encoding. XML-RPC is designed to be as simple as possible, while allowing complex data structures to be transmitted, processed and returned.

1.4 Thesis Structure

In This Chapter we presented the need, aim and the integration between Web and Mobile Application. Next Chapters go as follow:

Chapter Two: Methodology has been presented Agile Methodologies, Scrum, meetings, Stories and Testing.

Chapter Three: Technologies have been presented open Sources in Web and Mobile components.

Chapter Four: Web Components have been presented introduction, user requirements, websites back end design, interface Design, implementation and testing.

Chapter Five: Mobile components have been presented introduction, user requirement, user interaction with mobile App and Simple code.

Chapter Six: Design components have been presented introduction, logo design, mobile icons and site layout.

Chapter Seven: Integration between web and Mobile presents introduction, integration methods, XML RPC and code samples.

Chapter Eight: project outcomes present the output of "Eventati" representing as screen Shots of outcomes of this project.

Chapter Nine: presents Conclusion and Future Works that we will do in the future In sha Allah.

Chapter 2

Methodologies

2.1 Agile Methodologies

Every project must follow dedicated method to make a great project. Traditionally, Waterfall used as a methodology to develop a project, but waterfall has some limitations to be used until now, so agile methods will be used in almost project. In the Agile Method, communication between customers and the members of the project team is very vital and essential. Agile management in the project helps you defining the project easily and clearly with stakeholders and team members. The basic designs are kept simple and clean and feedback starts immediately as does software testing. This enables the project to be implemented early as changes are made immediately to any part of the project that requires it. The Agile method also allows the product to be launched after each stage and allows parallel analysis. In Contrast the waterfall that has no stopping and if end-testing doesn't go well, you better think about starting all over again. One of the leading agile software development processes is Scrum. Figure 2.1 presented in page 6 shows different between Waterfall and Scrum [flicker].

2.2 SCRUM

Scrum has become recognized as one of the best project management frameworks for handling rapidly changing or evolving projects. Especially useful on projects with lots of technology or requirements. It is a method for mapping software projects and product or application development. It has one of the best agile development practices using today. In any project we can get features request from customers, executives or even other team members. In Scrum features are written from the perspective of the end

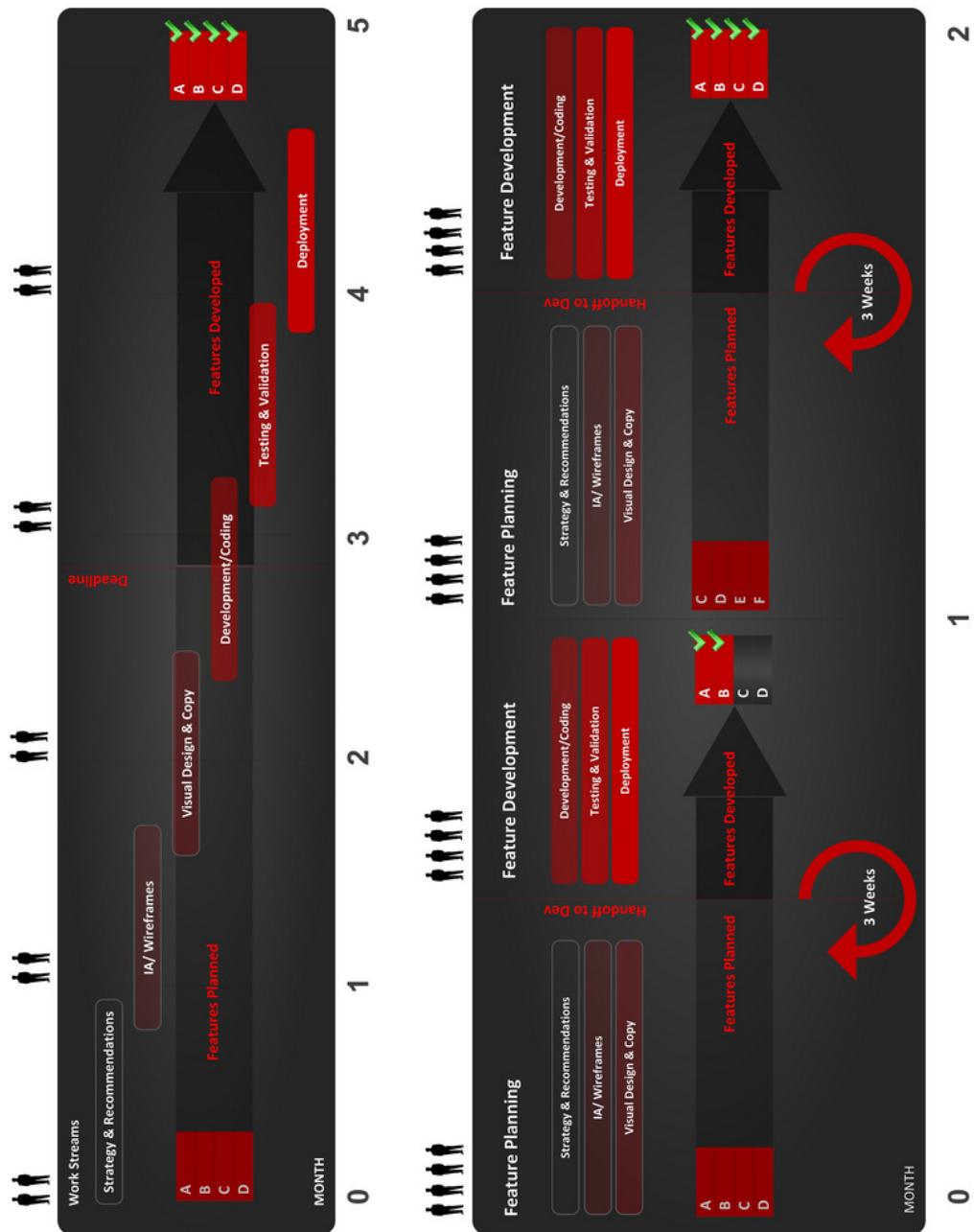


Figure 2.1: Waterfall and Scrum

user, therefore features are known as user stories, in collection of all these user stories are called product backlog. Product backlog is wish list of all things would make the product great. Once we have a product backlog or a wish list, we must determine which specific user stories we want to achieve first which has a priority. To build this product we need to have one or more people and has rules need to be followed:

1. Product owner - helps making sure the right features that represent user's requirements.
2. Scrum Master - to make sure the project is progressing smoothly and facilities release planning.
3. Developer - to build the project.
4. Tester - tested to make sure works rights
5. Customer - use it and helpfully to pay for it.

Scrum works with product Backlog which is nothing more than a list of features the weak high user's stories, then break down product backlog into one or more release backlogs and for agave release you further release backlogs into numbers of sprint backlogs and then your manner progressively sprint use Burn down Charts and have daily scrum meetings to ensure everything in on track after each sprint. We have a sprint retrospective to find everything [Hamid Shojaee]. Figure 2.2 presented in page 8 shows all the processes in SCRUM. To Apply Scrum on "Eventati" ,the Figure 2.3 presented in page 9 shows one of our scrum team members and Figure 2.4 presented in page 9 shows Product Backlog of "Eventati".

2.2.1 Meetings

According to scrum we made a lot of meetings almost 20 meetings all team meet together to complete and work for "Eventati" Project. We began used XMind tool beginning with Meeting5 and we had more than 15 meetings by the supervisor "Dr Haitham A.E-Ghareeb" and other meetings with team members only. Every meeting has been recorded with date,place,Start and end time , attendances,last tasks that we discussed and had to do this and new tasks that we will make them. For first five meetings didn't make it with XMind tool,then we reminded them shortly .

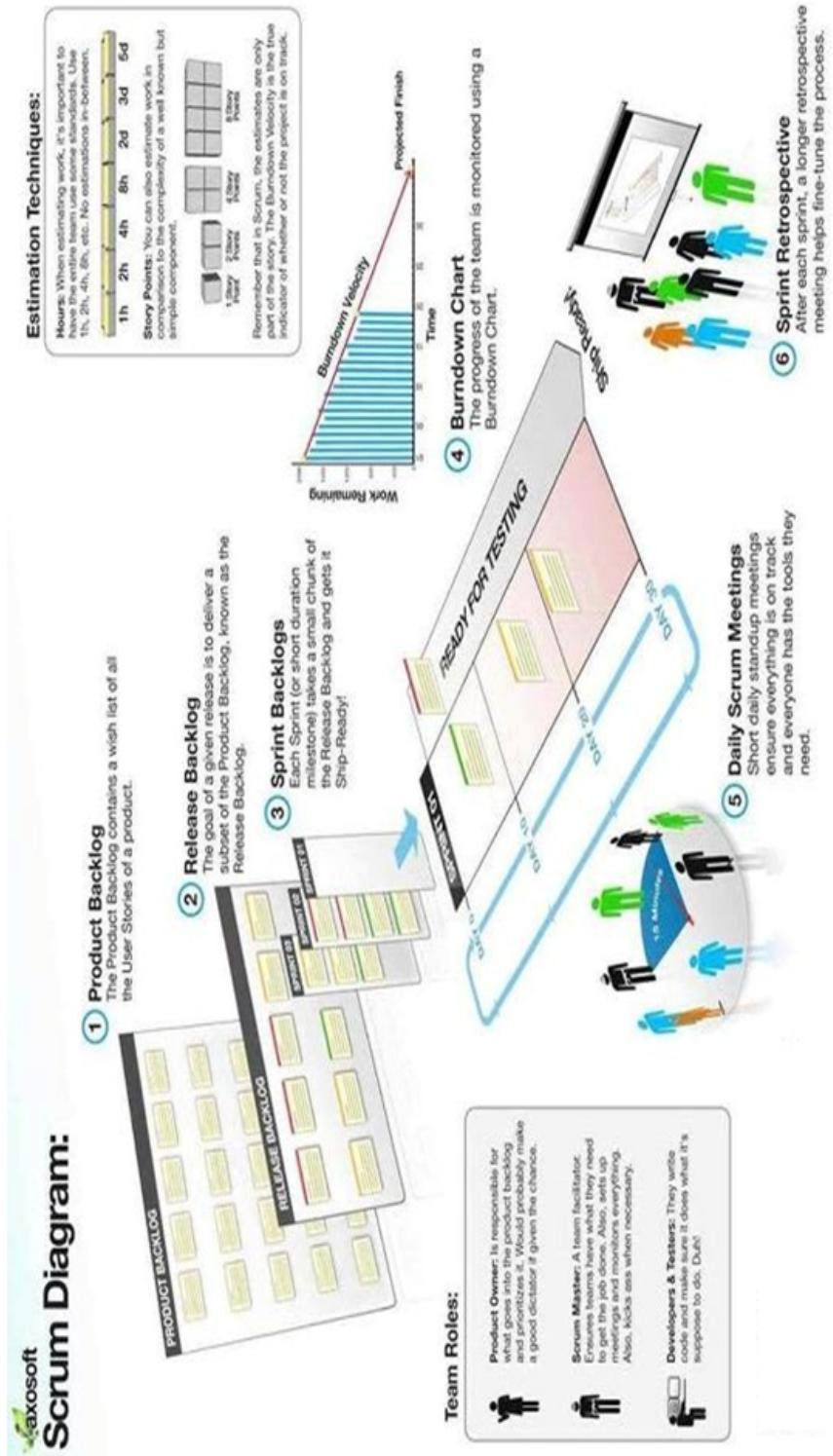


Figure 2.2: SCRUM Diagram



Figure 2.3: Scrum Team Members



Figure 2.4: Product Backlog Of "Eventati"

1. Meeting 1: we talked about project and what it's dimension then we revised java Script and discussed communication tool with others.
2. Meeting 2: we explained Scrum and divided the team into three major teams Web,Design and Mobile team and discussed MVC and language tool Web2by that the web team used it firstly.
3. Meeting 3: we talked about SVN , Google code, discussed the different types of Remote Execution Methods and standards: XML-RPC, SOAP (implements WSDL), Restful (implements WSDL), JSON,then Design team presented first work Mockups on paper that it showed System web site and all team impressed with their work they will update it with Cacoo tool next time .
4. Meeting 4: "Dr Haitham" meets with each team individually and discussed about tools and what each team will study and be suitable for the "Eventati" project .
5. Meeting 5: Place: ILC,date: 2-10-2010 at 10.45 pm We discussed that day Features of the project and determined weekly meeting for each team.

6. Meeting 6: Figure 2.5 presented in page 12 showed that discussion of three things. Firstly, we discussed analysis that we need Use case, Activity and sequence Diagrams. Secondly, we discussed the Test driven development (TDD) how to test, why should we first test, why should another one test, why test first is good. Thirdly, we discussed some issues that we have met, one of these is if an event is cancelled, it must be in updating not make deleting operation.
7. Meeting 7:Figure 2.6 presented in page 13 showed that according to Mobile Team we discussed about titanium that we used it to build "Eventati" mobile application and distributed tasks on team, Web team decided using Django instead of using web2by because it was easy and fast in development and performance otherwise web2by was not customized and need URL function Mobile and discussed these tasks Install the environment, subversive, Events Manager to be finished and user Registration using layout.
8. Meeting 8:Figure 2.7 presented in page 14 showed that it determined Events entities, source Code management, Product Backlog version 1 and suing "yamel" tool for design team.
9. Meeting 9:Figure 2.8 presented in page 15 showed that we discussed Logo rules and new technologies that we used like pianx and bootstrap then mobile team needed icons and discussed about presentation and Documentation.

These teams with out "DR.Haitham" the supervisor:

1. Meeting 1:Figure 2.9 presented in page 16 showed that the first team with out our supervisor "Dr Haitham" that we rearranged new tasks and distributed them on all team members. UML, Attributes, Icons and determined tasks to next meeting we needed a Survey, Class, Sequence, Activity Diagram and complete home page (Web and Design Team).
2. Meeting 2:Figure 2.10 presented in page 17 showed the initial diagrams Activity, sequence and class Diagrams and we needed to revise it with "Dr Haitham".
3. Meeting 4:Figure 2.11 presented in page 18 showed that we discussed Main, Home pages, Logan and Slogan .we had notes to be discussed with "Dr Haitham".
4. Meeting 5:Figure 2.12 presented in page 19 showed that we discussed Video Or slide Show to represent our system site to users ,then we

discussed Slogan of "Eventati" that every member in team suggested at least two slogan and at the end we voted the slogan "Every Moments Something Happens" that took the highest votes.

The Last Two Meeting be collected every things that we done with out "Dr Haitham" to Show them to him.

1. Meeting 10:Figure 2.13 presented in page 20 showed that this meeting was with "DR Haitham" discussed presentation according to with design team about fonts and position of slogan and logo and rearranged of slides.According to mobile team that they needed a vital pictures not Screen shoots and need a spacial Use Case and Diagrams.According to web team that we needed statistical for presentation,MVC Just one Slide,Benefits of RSVP need to updated.
2. Meeting 11:Figure 2.14 presented in page 21 showed that this also was with "DR Haitham" there was final updating to this presentation of "Eventai",Use Case Diagrams and talked about how to represent it.
Great Thanks To "DR Haitham" for helping us too much and not saving his efforts at all.

2.2.2 Stories:

Stockholders write user stories; user stories mean features of our project. User Stories of "Eventati":

- User can create Events.
- User can invite friends.
- User can publish Events through Mail, Social network or "Eventati".
- User can get location of each event and friend.
- User can follow new friends.
- User can rate each event.
- User can leave a comment in each event.
- User can confirm attending with RSVP.
- System can notify all subscribes.
- System can recommend places, friends, events to Authorized users

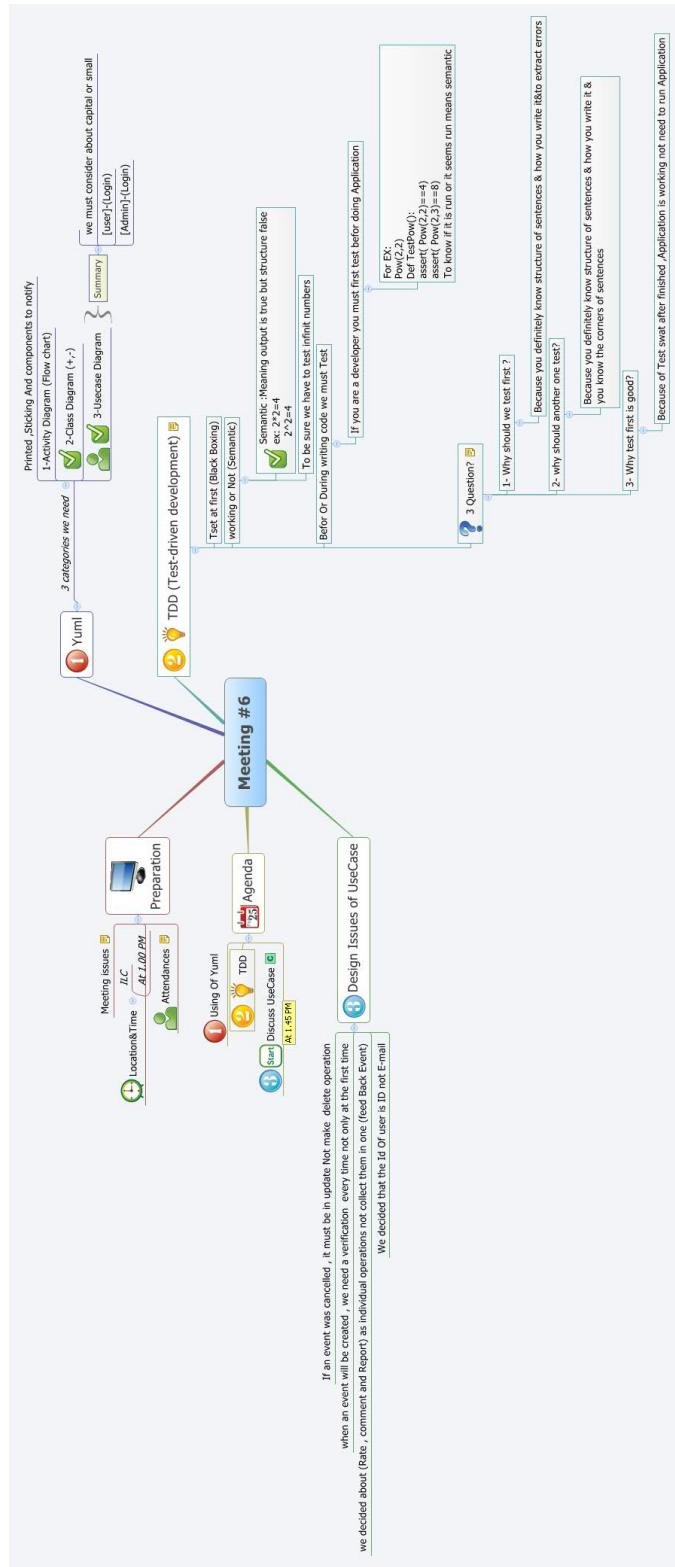


Figure 2.5: Meeting 6
12

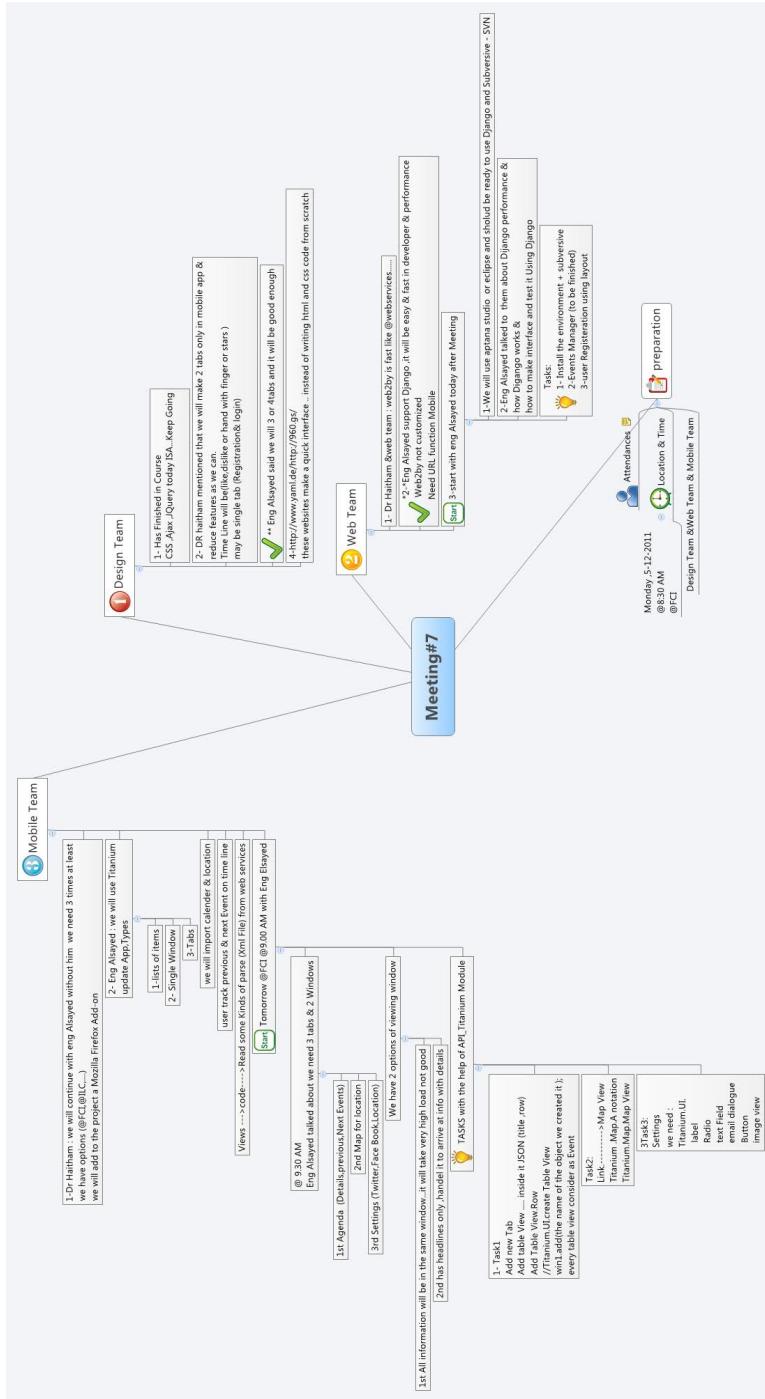


Figure 2.6: Meeting 7

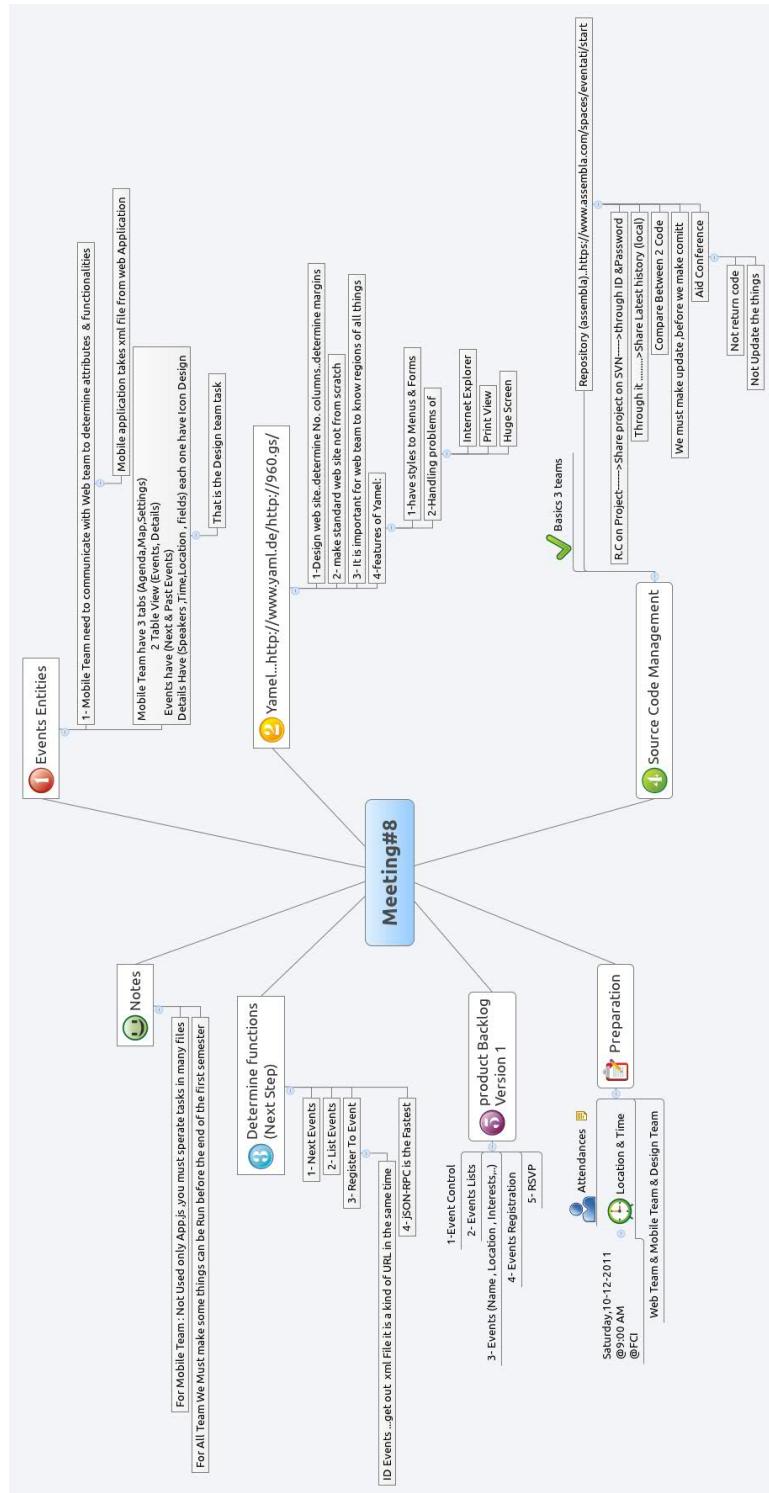


Figure 2.7: Meeting 8

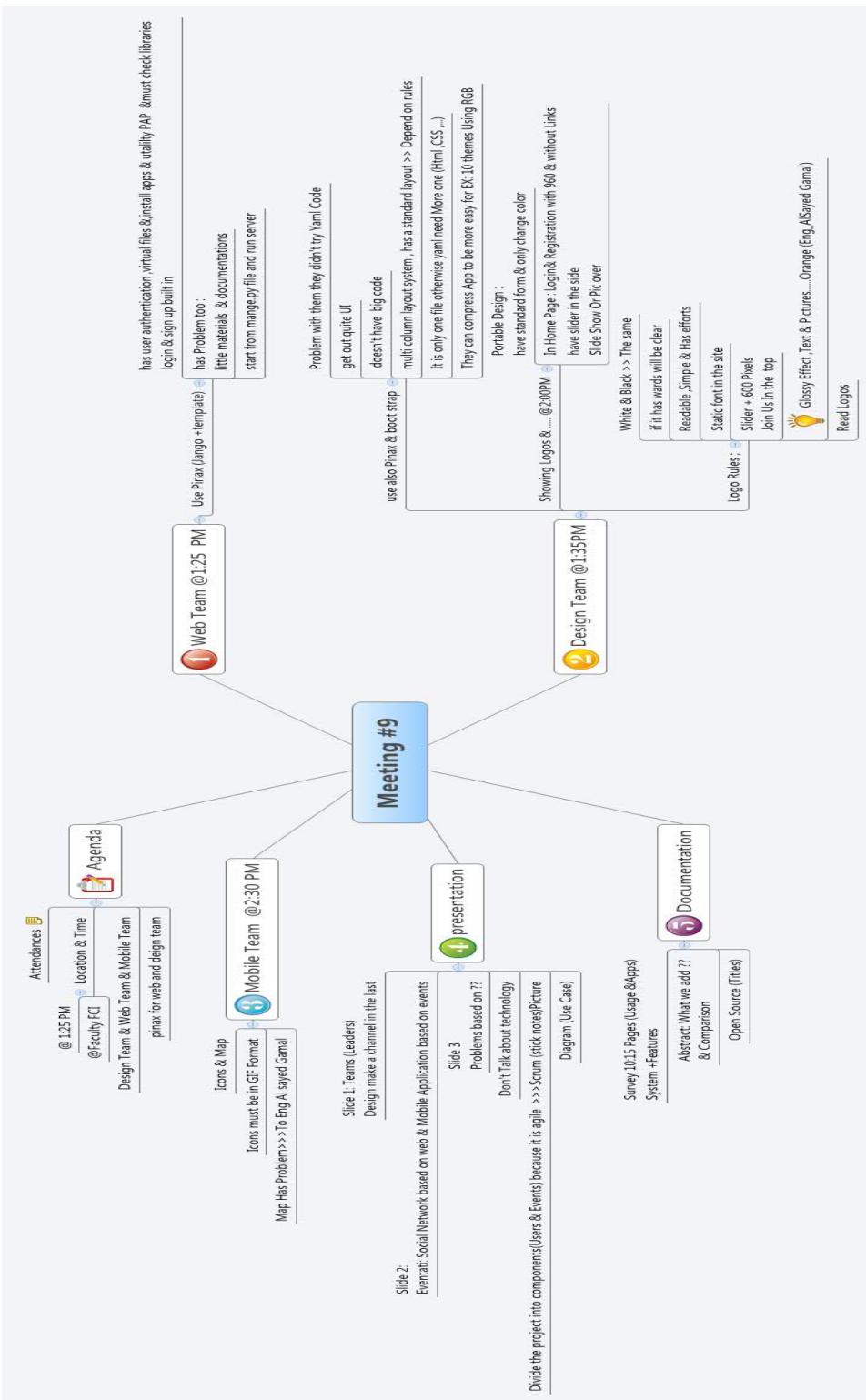


Figure 2.8: Meeting 9

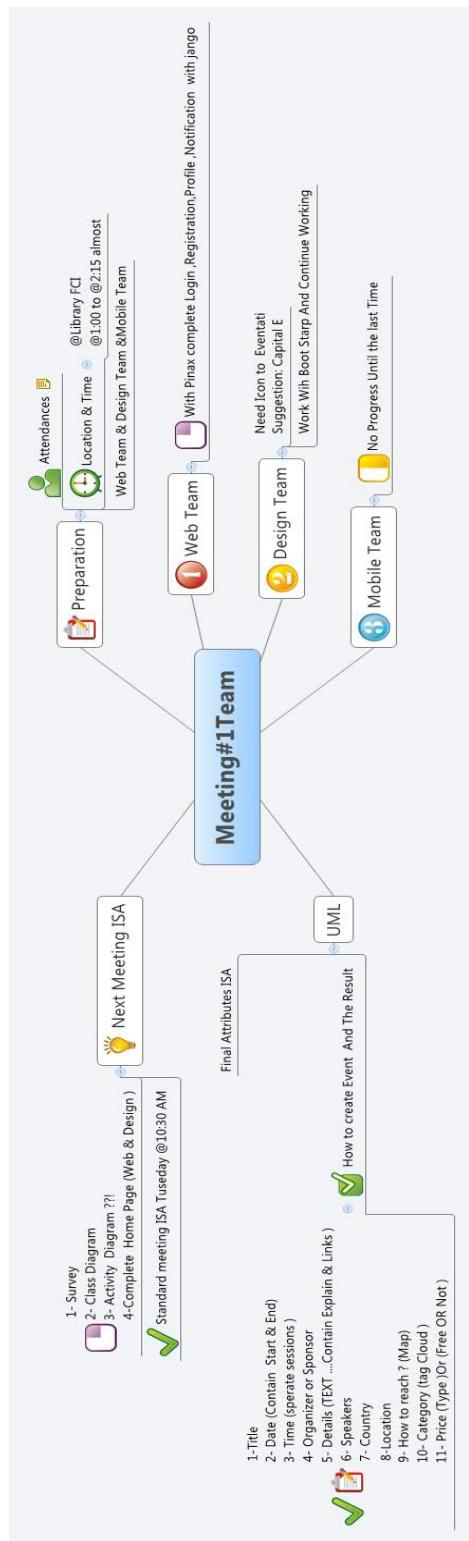


Figure 2.9: Meeting 1 Only Team

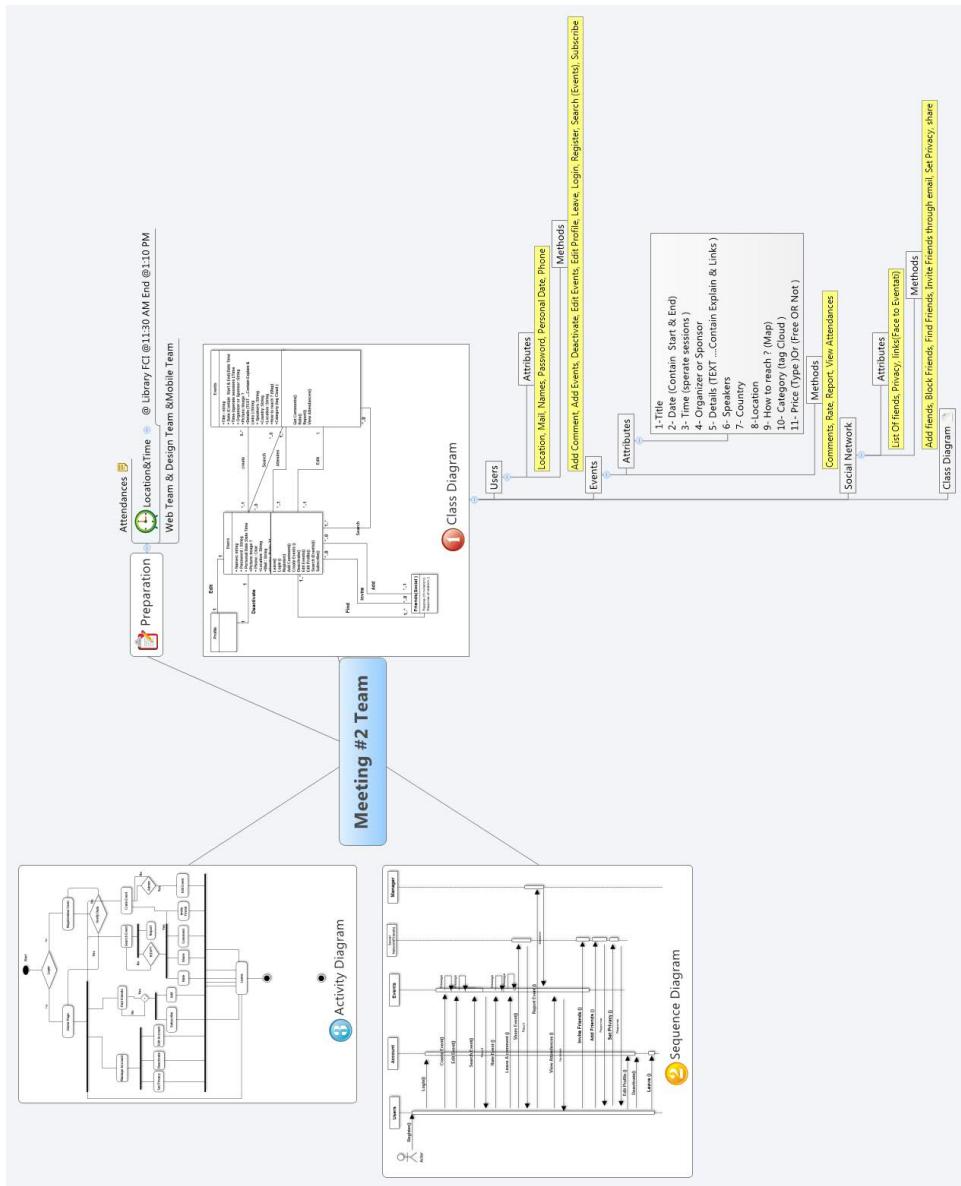
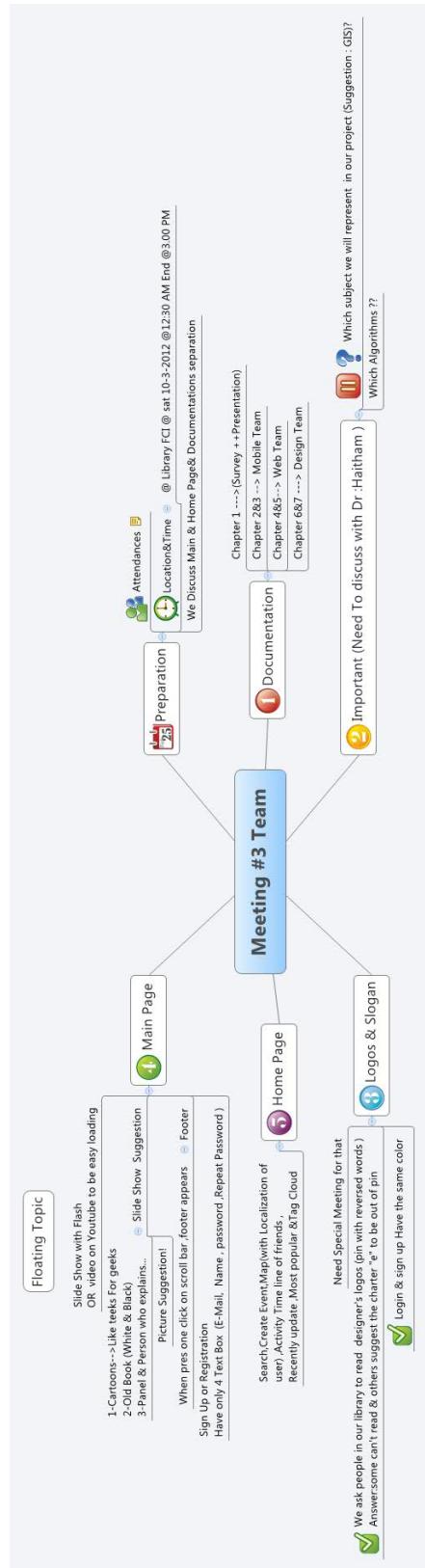


Figure 2.10: Meeting 2 only Team



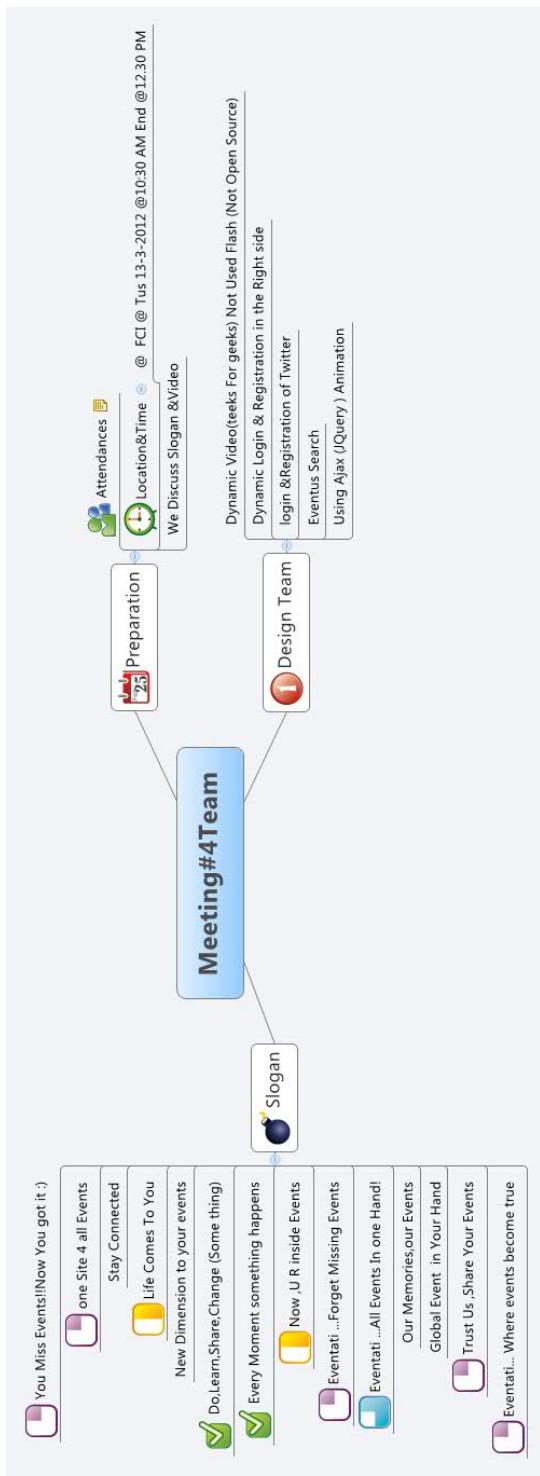


Figure 2.12: Meeting 4 only Team

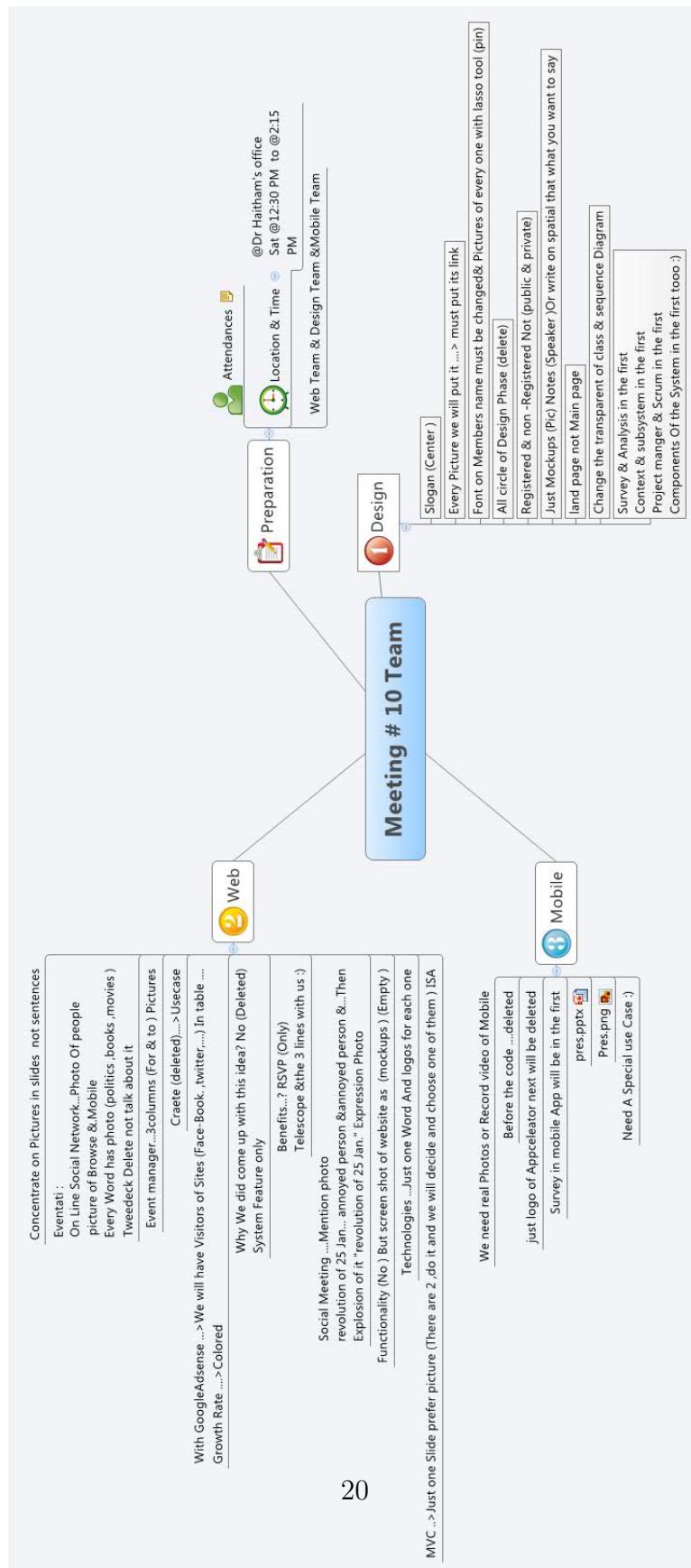


Figure 2.13: Meeting 10

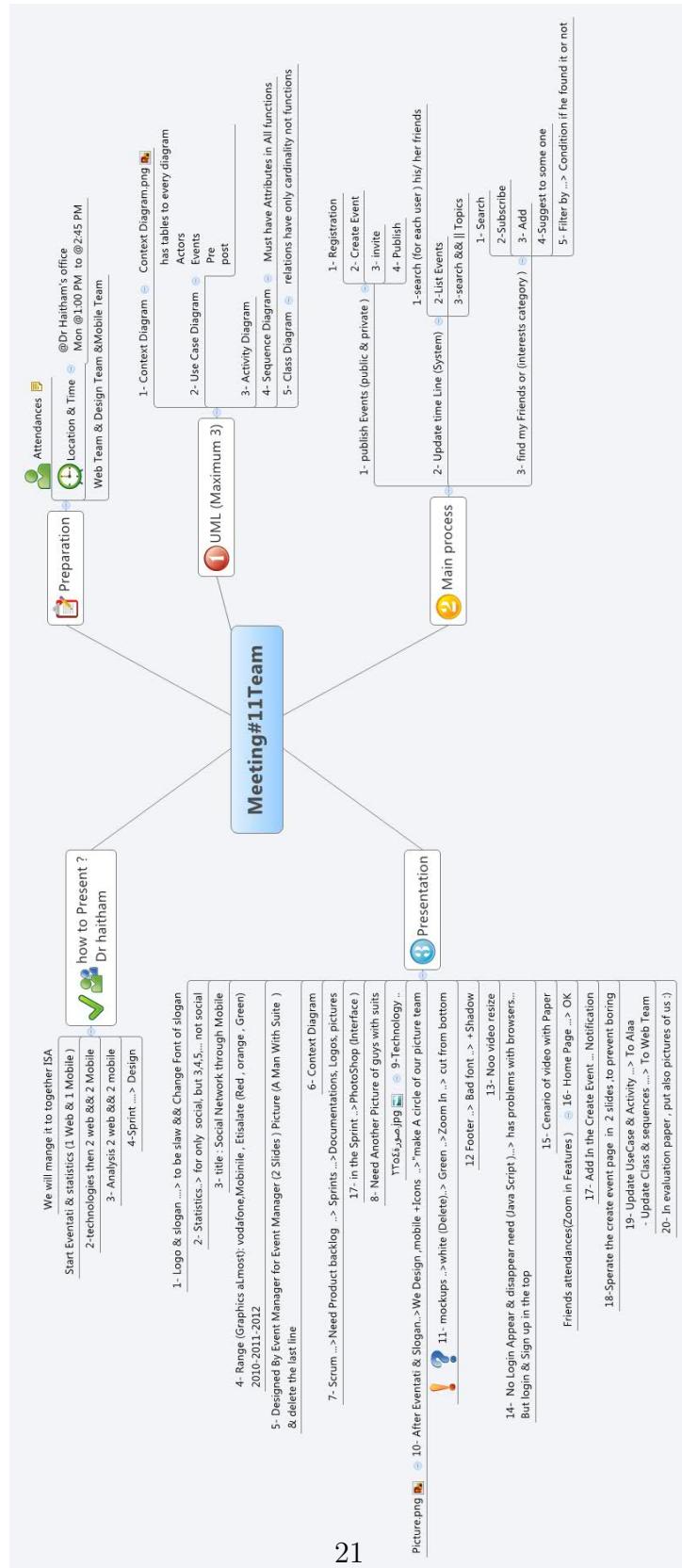


Figure 2.14: Meeting 11

2.2.3 Testing:

When our software is completing , it must be tested against the system requirements to see if it fits the original goals. As well as this kind of conformance testing, it's a good idea to see if our software can be broken via its external interfaces , this helps to protect us against accidental or malicious abuse of the system when it's been deployed.

Testing is intended to show that a program does what it is intended do and to discover program defects before it is put into use. We need testing to validate, defect Verify and validate our project. Other important rules we need to differentiate between inspections and testing. Software Inspection is Static Verification while Software Testing is dynamic Verification. There are many different types of testing. The prominent ones this series will cover are unit tests and integration tests.

Another point of decision is deciding whether to do test-first or test-after. Test-first is where you write the necessary tests to demonstrate proper behaviour of the code BEFORE you write the code to solve the problem at hand. Test-after is when you've already written the code to solve the problem, then you go back and create tests to make sure the behaviour of the code you wrote is correct.

Chapter 3

Technologies

3.1 Web Technology

In web application we used major open source technologies python, Django framework, and pinax you can take some information about them in following lines.

3.1.1 Python

Python is an interpreter, object-oriented and high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding; make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is



Figure 3.1: Python Logo



Figure 3.2: Django Logo

incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective [Python].

3.1.2 Django

Because of Django was developed in a fast-paced newsroom environment, it was designed to make common Web development tasks fast and easy .Django is an open source web application framework, written in Python, which follows the model-view-controller architectural pattern. It was originally developed to manage several news-oriented sites for The World Company of special place in USA, and was released publicly under a BSD license in July 2005; the framework was named after guitarist Django Reinhardt. In June 2008 it was announced that a newly formed Django Software Foundation will maintain Django in the future. Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes reusability and "pluggability" of components, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings, files, and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models[Adrian Holovaty and Jacob Kaplan-Moss]. Django Application:

Django is a free web framework written in python has its own typical structure

1. You have an application folder "apps", where all your application are.
2. In the static folder you have all static files like, css, js, images etc..
3. In custom folder we put our own logic with we overwrite some specific application.
4. In template folder we put our templates, how the website should look like.

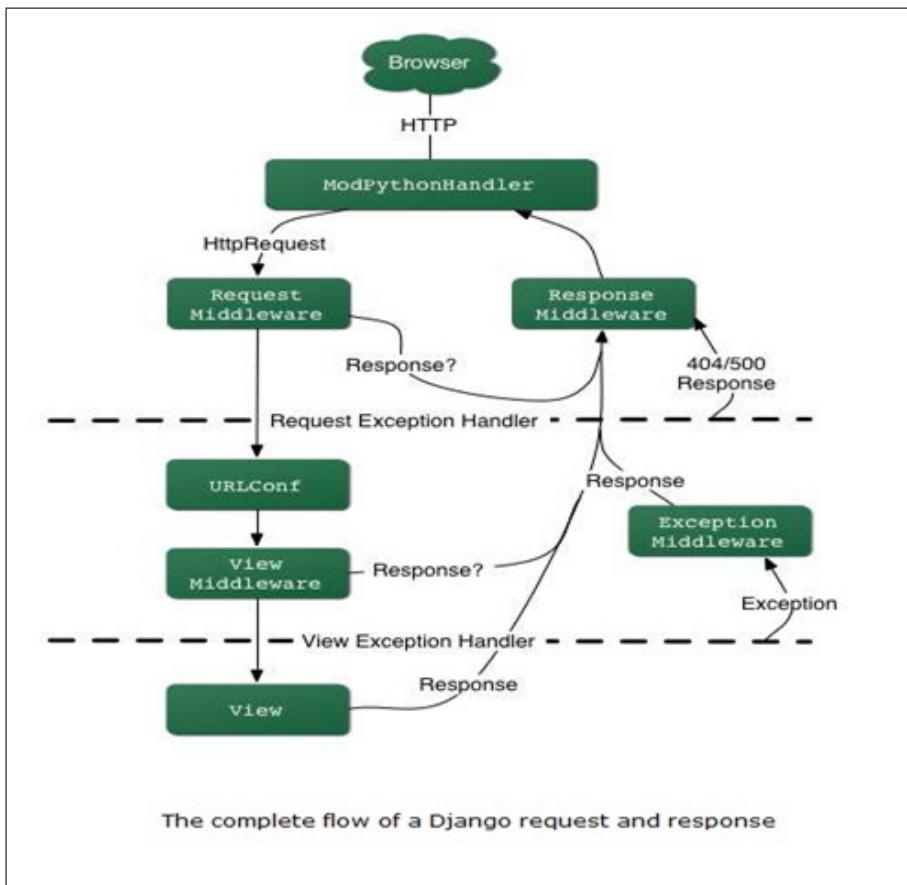


Figure 3.3: Django Request and Response

5. The files in the root project tree, like settings and urls.py are global for the project.

This is a common structure, but you can make it however you want, this is not a requirement. In Django you can overwrite most of the behaviour [Django Application] and [by book keeper].

3.1.3 The MVC Design Pattern

Let's dive in with a quick example that demonstrates MVC, The first thing to note is that we split it over three Python files (models.py, views.py, urls.py) and an HTML template latest-books.html. models.py (the database tables), views.py (the business logic), urls.py (the URL configuration) and latest-books.html (the template).

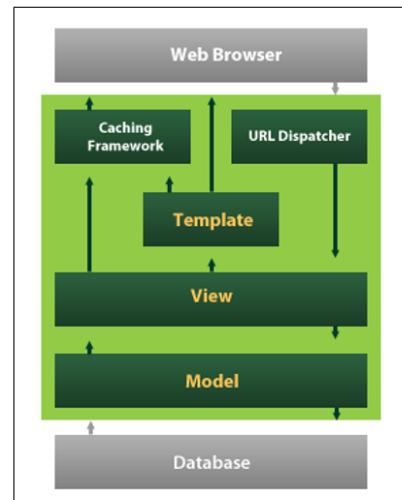


Figure 3.4: Django Application

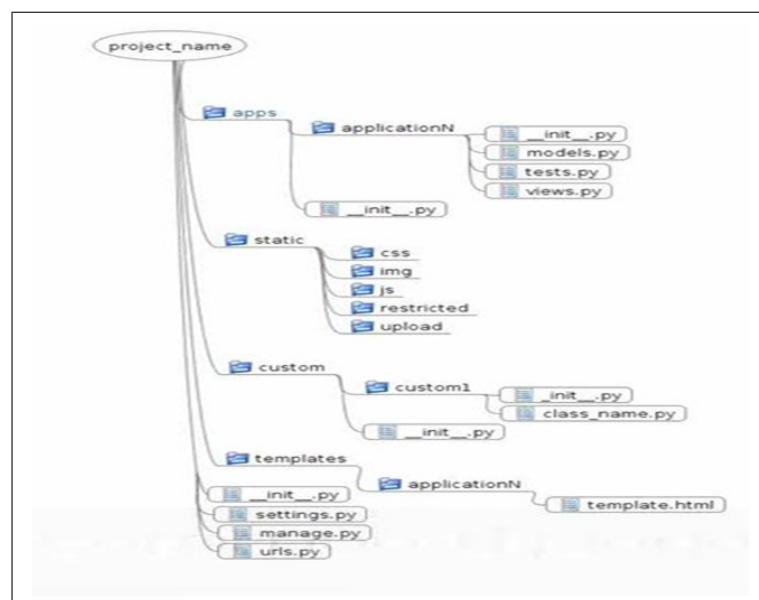


Figure 3.5: Django Application Structure

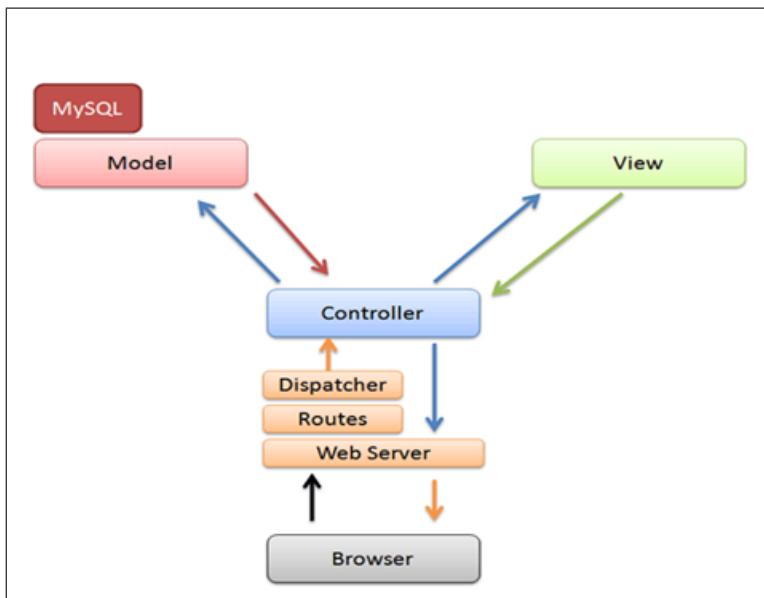


Figure 3.6: MVC Design Pattern

```

1  from django.db import models
2  class Book(models.Model):
3      name = models.CharField(max_length=50)
4      pub_date = models.DateField()
5
6
7
8
9

```

Figure 3.7: models.py (the database tables)

- The `models.py` file contains a description of the database table, represented by a Python class. This class is called a model. Using it, you can create, retrieve, update, and delete records in your database using simple Python code rather than writing repetitive SQL statements.
- The `views.py` file contains the business logic for the page. The `latest_books()` function is called a view.
- The `urls.py` file specifies which view is called for a given URL pattern. In this case, the URL `/latest/` will be handled by the `latest_books()`

```
 1
 2 # views.py (the business logic)
 3 from django.shortcuts import render_to_response
 4 from models import Book
 5 def latest_books(request):
 6     book_list = Book.objects.order_by('-pub_date')[:10]
 7     return render_to_response('latest_books.html', {'book_list': book_list})
 8
 9
```

Figure 3.8: views.py (the business logic)

```
 1
 2 # urls.py (the URL configuration)
 3 from django.conf.urls.defaults import *
 4 import views
 5 urlpatterns = patterns('',
 6     (r'^latest/$', views.latest_books),
 7 )
 8
 9 6 Chapter 1 ? INTRODUCTION TO DJANGO
10
11
```

Figure 3.9: urls.py (the URL configuration)

```
 1
 2
 3 # latest_books.html (the template)
 4 <html><head><title>Books</title></head>
 5 <body>
 6     <h1>Books</h1>
 7     <ul>
 8         {% for book in book_list %}
 9             <li>{{ book.name }}</li>
10         {% endfor %}
11     </ul>
12 </body></html>
13
```

Figure 3.10: latest-events.html (the template)



Figure 3.11: pinax

function. In other words, if your domain is example.com, any visit to the URL [<http://example.com/latest/>] will call the latest-books() function.

The latest-books.html file is an HTML template that describes the design of the page. It uses a template language with basic logic statements for example, Taken together, these pieces loosely follow a pattern called Model-View-Controller (MVC).Simply put, MVC is way of developing software so that the code for defining and access in data (the model) is separate from request-routing logic (the controller), which in turn is separate from the user interface (the view). A key advantage of such an approach is that components are loosely coupled. Each distinct piece of a Django-powered Web application has a single key purpose and can be changed independently without affecting the other pieces. For example, a developer can change the URL for a given part of the application without affecting the underlying implementation. A designer can change a page's HTML without having to touch the Python code that renders it.A database administrator can rename a database table and specify the change in a single place rather than having to search and replace through a dozen files.

3.1.4 Pinax

A platform for rapidly developing websites Pinax is an open-source platform built on the Django Web Framework. By integrating numerous reusable Django application and providing starter projects and infrastructure tools, Pinax takes care of the things that many sites have in common so you can focus on what makes your site different. Pinax has been used for everything from social networks to conference websites, and from intranets to on-line games. Pinax provides:

- standard project layout for consistency and easy deployment

- starter projects that can be used as the basis for any Django website as well as some tailored to community sites, company sites, intranets and sites in closed beta
- reusable applications providing both back-end functionality and user-facing components
- default templates to enable quick prototyping including applications for:
 - account management
 - open id
 - e-mail verification
 - password management
- profiles
- notifications
- activity streams
- private betas / waiting lists
- tagging
- wikis

[James Tauber]



Figure 3.12: Appcelerator Logo



Figure 3.13: Titanium Logo

3.2 Mobile Technology

3.2.1 Appcelerator

”Eventati” Mobile team used open source technologies,because of open source is better quality, higher reliability, more flexibility and lower cost. There are many problem that we faced and overcame with titanium,problems such as [Appcelerator,Inc]:

- Building Cross Platform Mobile Solutions.
- Building Applications in cost-effective manner.
- Providing best of breed user Experience.
- Building a sustainable, maintainable for application and team .

Appcelerator is an open source developer platform designed to allow programmers to create native applications that function across a wide range of devices.Appcelerator Titanium Mobile is one of several phone web based application framework solutions allowing web developers to apply existing skills to create native applications for iPhone and Android. Yet, while using the familiar JavaScript syntax, developers will also have to learn the Titanium API[Richard Bate-man]. The language that ”Eventati” mobile team used Appcelera-tor==JS+TitaniumAPIs We used titanium with java script,write code in JSON. JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the

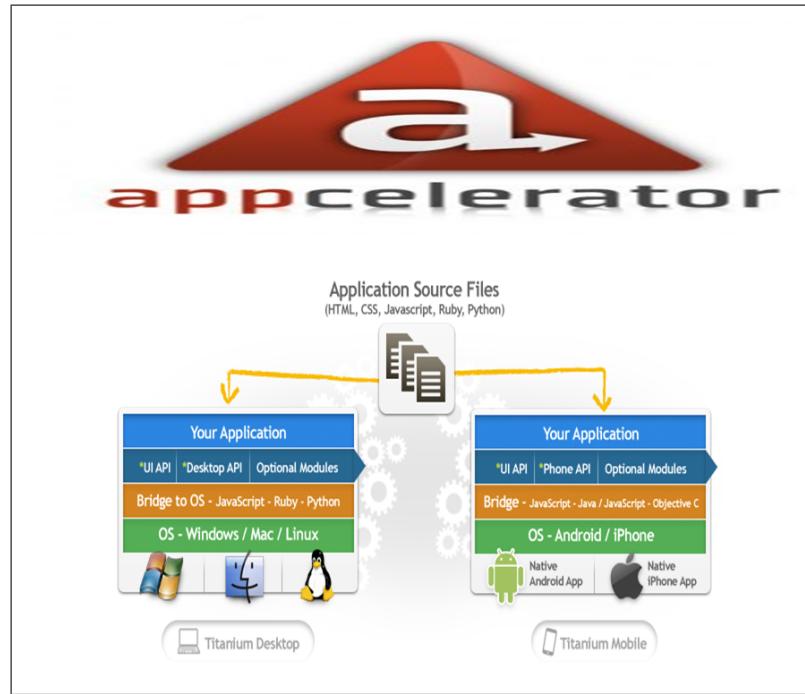


Figure 3.14: Appcelerator of Application Sources File

JavaScript Programming Language. The code gets compiled using the device SDKs so the deployed application runs natively on the phone not only android but also IOS.

Today you need to be in three places at once: On-line, On-phone, and On-desktop The html/css/javascript code that makes up the core application logic and UI, All of this can be done through Appcelerator Titanium [Viktor, ShoutEm CEO].

Chapter 4

Web Components

4.1 Introduction

An API, or “ Application Program Interface, is a set of routines and protocols that provide building blocks for computer programmers and web developers to build software applications. In the past, APIs were largely associated with computer operating systems and desktop applications. In recent years though, we have seen the emergence of web API. Website development is an extensive term for the work involved in developing a web site for the World Wide Web. Web development services can range from developing a simple static web page of plain content to the most complex web-based internet applications, e-commerce businesses, or social network services. Moreover, for larger organizations and businesses, web development teams can consist of hundreds of web developers, whereas smaller organizations may only require a single web master and a secondary member assigned to related job positions such as a graphic design or a server testing. Web development is a collaborative effort between various departments rather than the domain of a single specific department.

4.2 System Requirements

4.2.1 Publish Event

Registration on the web application is the first functional requirement of our system after it come other requirement some of them are functional and another are Non-functional, we will make a trip in our system and specify it's requirement (functional,Non-functional). After registration the system gives to user an optional requirement it is completing user's profile it isn't



Figure 4.1: System Requirements

barrier for the user to do other process of the system. After that user should create his event and try to publish his event via many different ways such as share event via "Eventati" network or other social network and after this the event will be appear to the user friend in all network s/he shared his event on it and he will receive notification about his friend acceptance and joining his event all this process only for registered user. Figure 4.1 presented in page 34 shows how to get System Requirements. "Eventati" Consists of four components user management, Event management, Social Network and Mobile Application shown in Figure 4.2 presented in page 35 shows Context Diagram. Figure 4.3 presented in page 36 shows Web-UseCase-PublishEvent.

Table 4-1 description Of Publish Events.

Figure 4.4 presented in page 37 shows Web-Activity-PublishEvent.

Figure 4.5 presented in page 38 shows Web-Sequence-PublishEvent.

4.2.2 Find My friends

After the previous process system will recommend friends for you if you want and this process is an optional on the system and you can consider

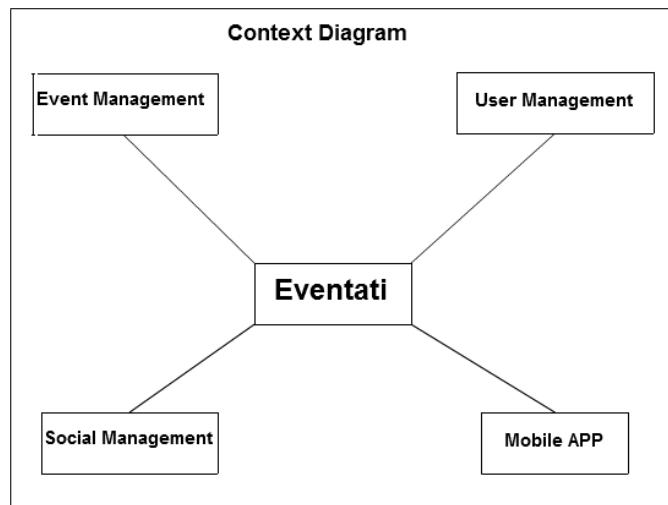


Figure 4.2: Context Diagram

Actor	User (Authorized People)
Description	An Authorized People can publish events via mail, Social Networks or "Eventati: ,Create Events, Search Events, Get Localization and invite Friends .
Data	Event's Attributes .
Stimulus	The Authorized send the data.
Response	Confirmation of the event has been published .
Comments	User must be authorized to make this process.

Table 4.1: Description of Publish Event Process

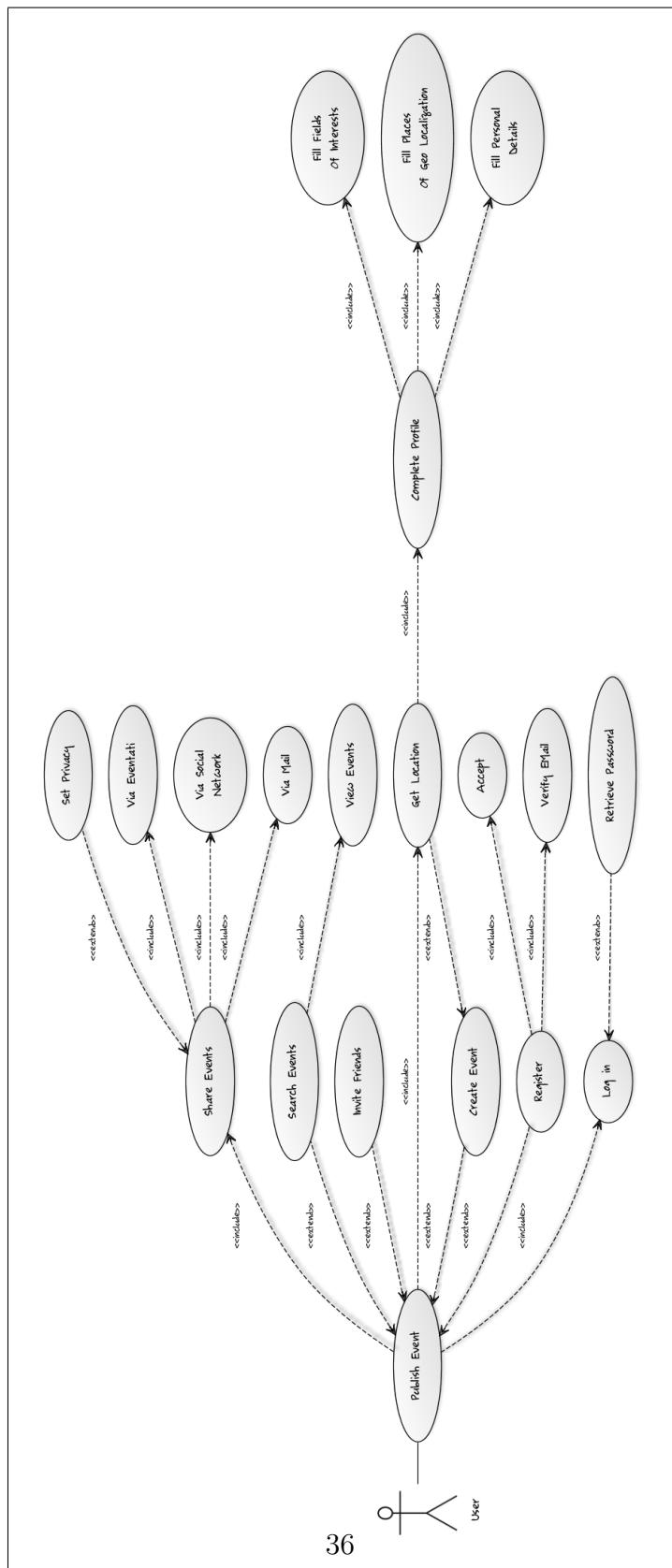


Figure 4.3: Web-UseCase-PublishEvent

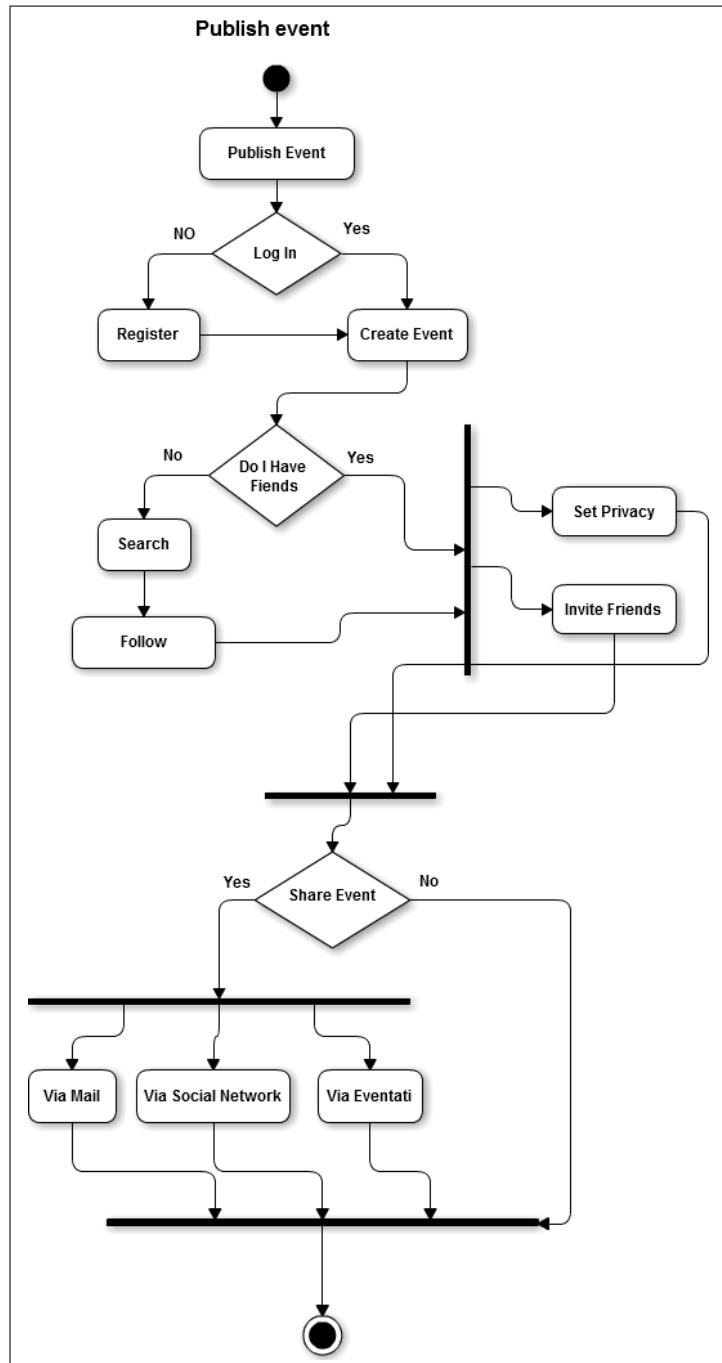


Figure 4.4: Web-Activity-PublishEvent

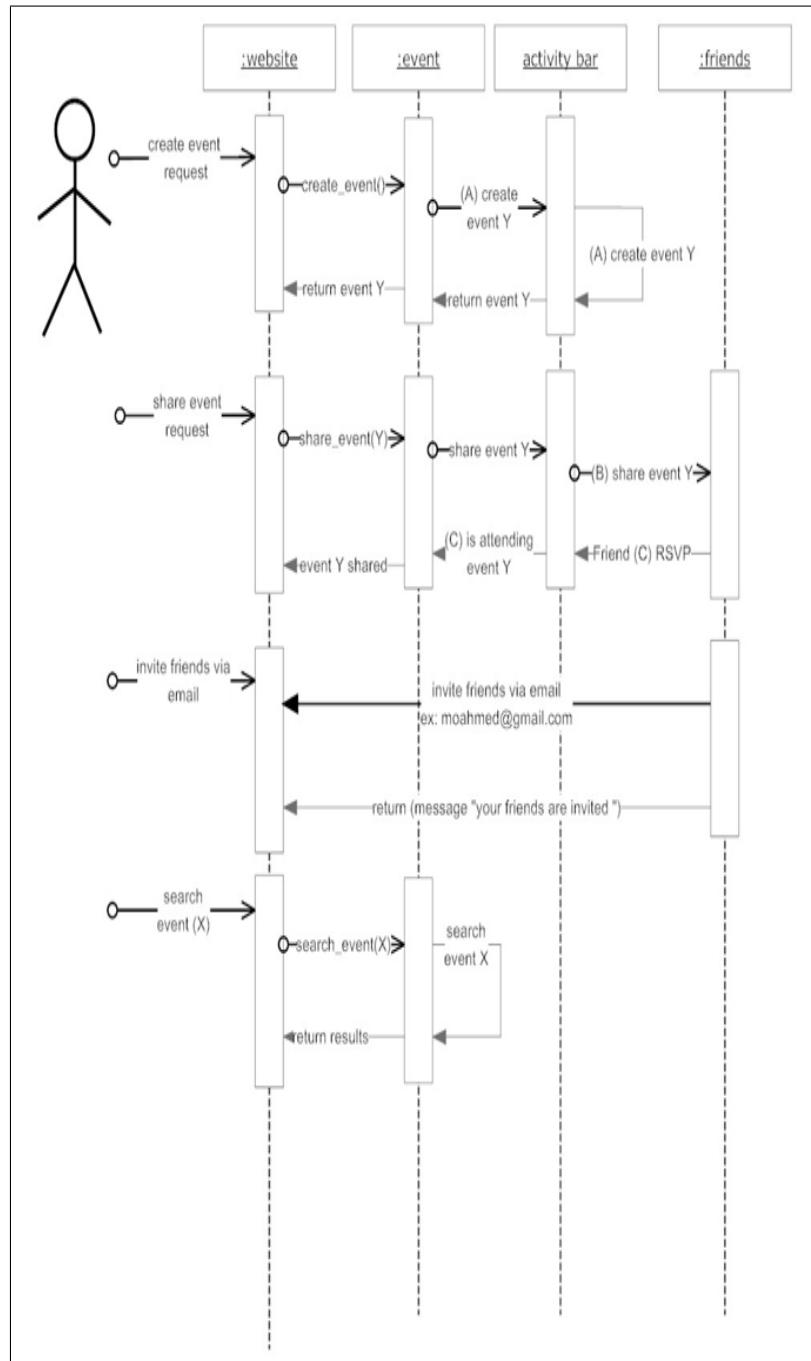


Figure 4.5: Web-Sequence-PublishEvent

Actor	User (Authorized People)and System
Description	An Authorized user can rate, leave a comment and confirm attending with RSVP to an event By Contrast System can notify all subscribers those happened, can also recommend events, friends, places and other activities that appear all in time line.
Data	Information that makes time line be updated or changed.
Stimulus	User command issued by authorized people and system .
Response	Confirmation or changing that happens on-line .
Comments	System must integrate with authorized user.

Table 4.2: Description of Find My Friends Process

it non-functional requirement of the system, another process of our system it is search that is very important process that system must do for the user (functional requirement of the system) search will on event and friends on the system. Also systems enable us to manage our account such as set privacy, change password and add phone number see. Find friend is one of most important activity for user on system it is available only for authenticated user who only can make this process. Figure 4.6 presented in page 40 shows Web-Use Case-Find My Friends Table 4-2 shows description Of Find My friends Process .And

Figure 4.7 presented in page 41 shows Web-Activity-Find My Friends. Figure 4.8 presented in page 42 shows Web-Sequence-Find My Friends

4.2.3 Update TimeLine

System can also make other functional requirements such as system update time line and it must be notified all users about last updates. It can also recommend activates, places for subscribers and also events. update time line such as confirmed attending via RSVP, left comments on events and rated events. Also system should provide some assurance technique for user to retrieve any personal data lost on system such as (password). Figure 4.9 presented in page 43 shows Web-Use Case-Update TimeLine Table 4-3 description Of Update Time Line Figure 4.10 presented in page 44 shows Web-Activity-Update TimeLine Figure 4.11 presented in page 45 shows Web-Sequence-Update TimeLine

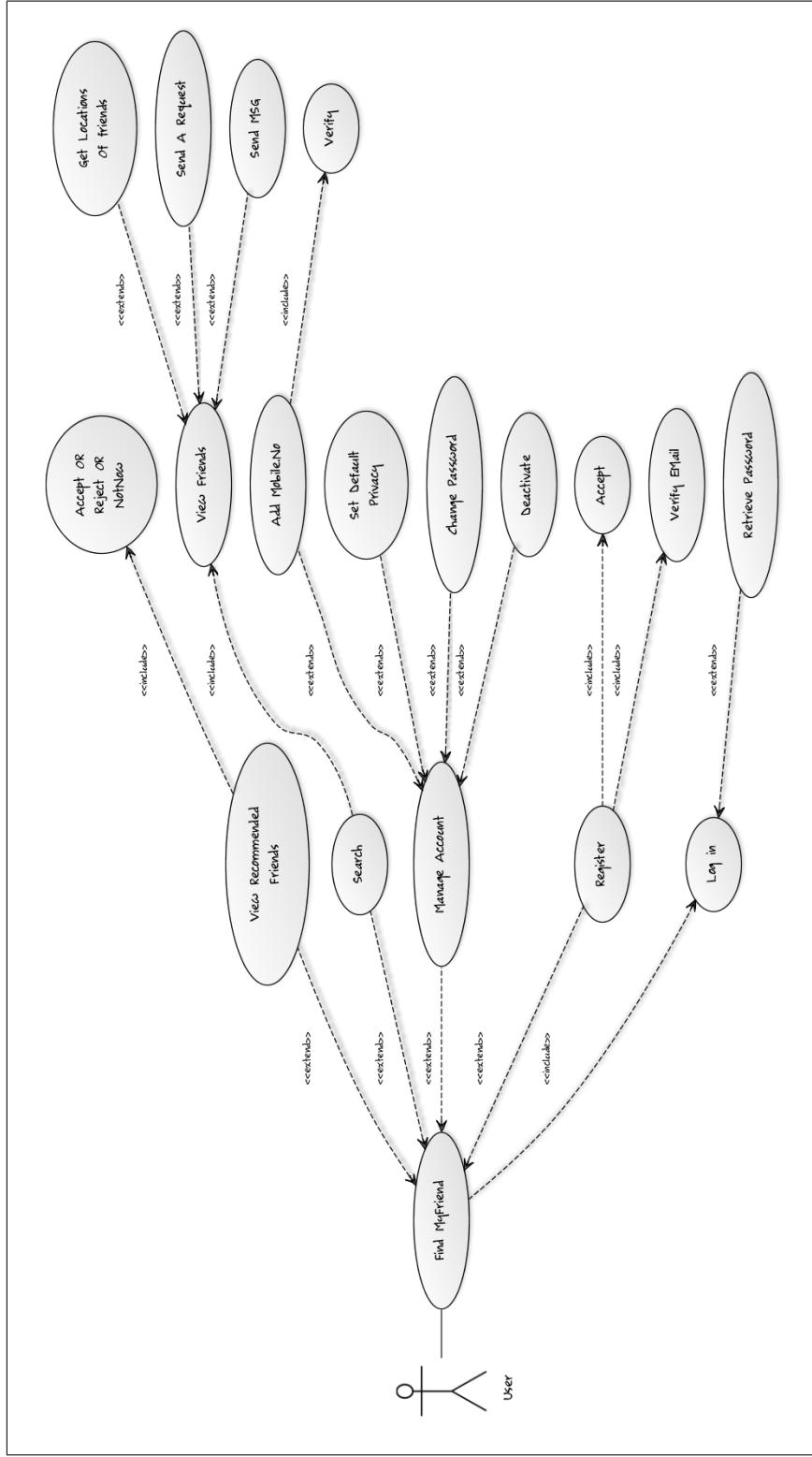


Figure 4.6: Web-Use Case-Find My Friends

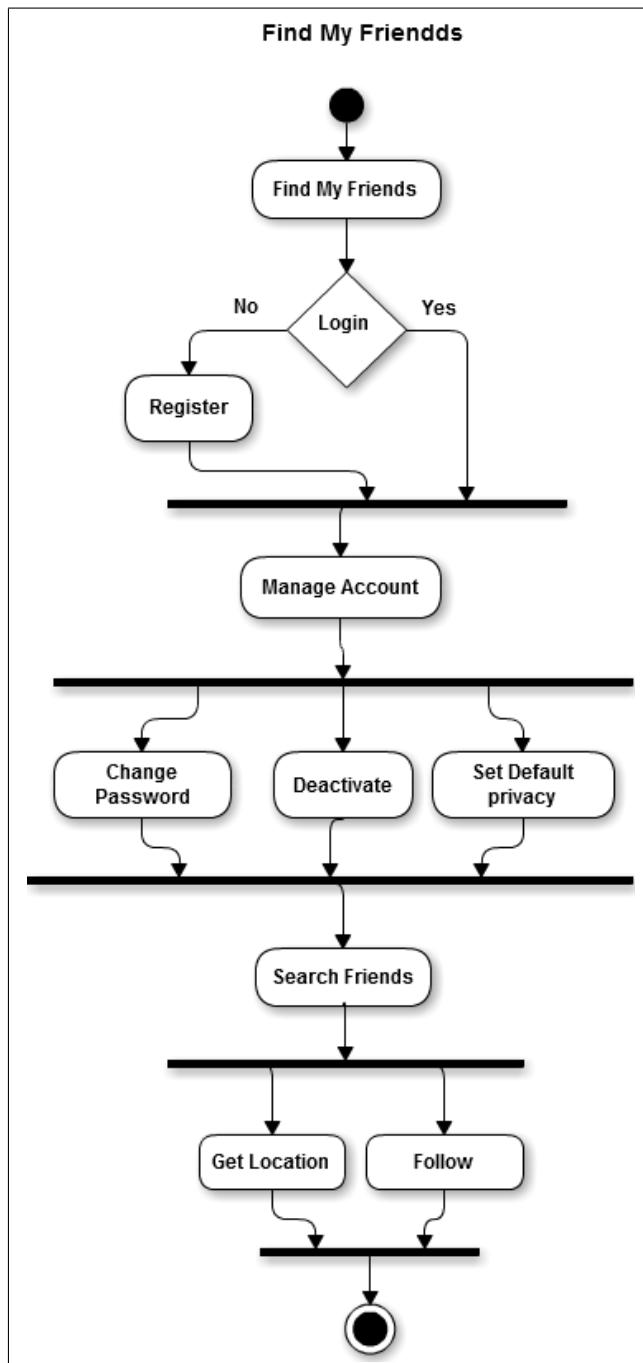


Figure 4.7: Web-Activity-Find My Friends

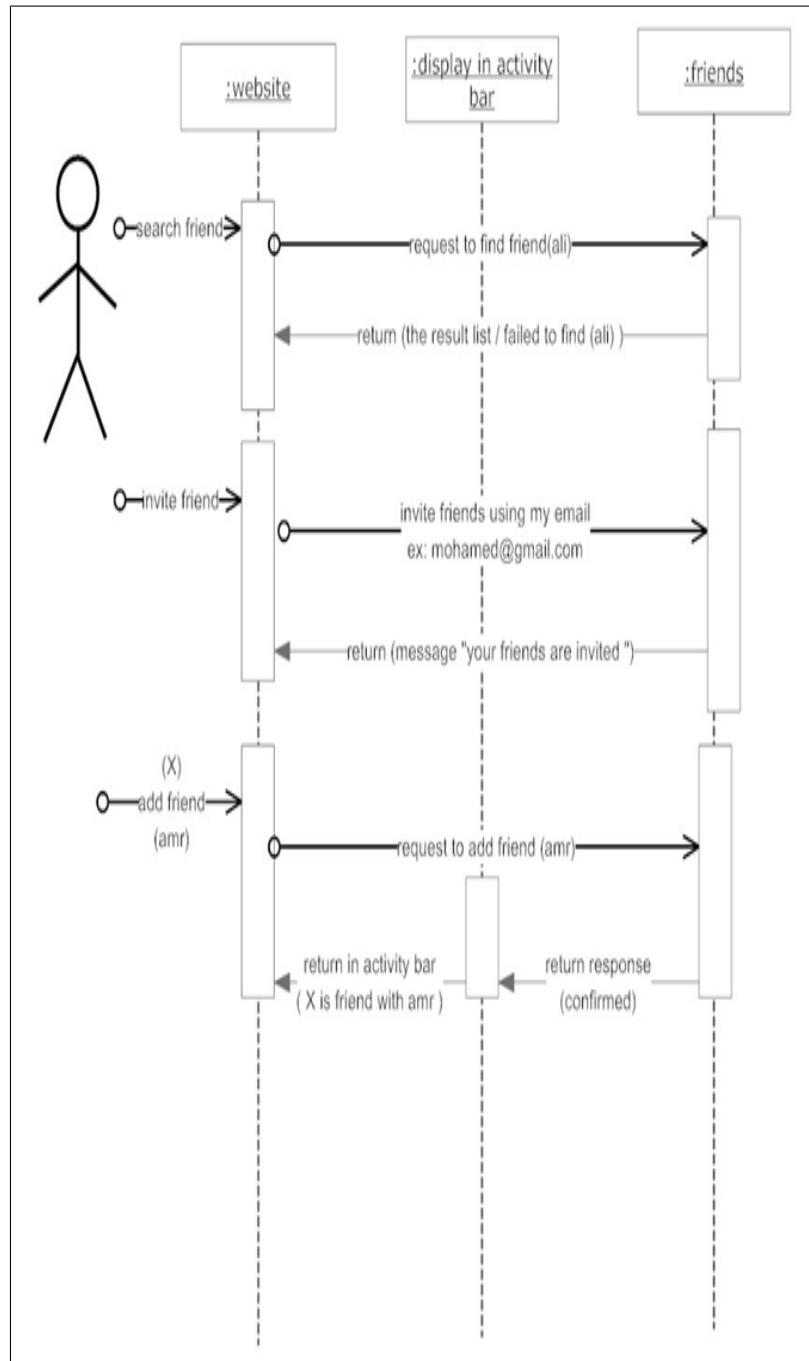


Figure 4.8: Web-Sequence-Find My Friends

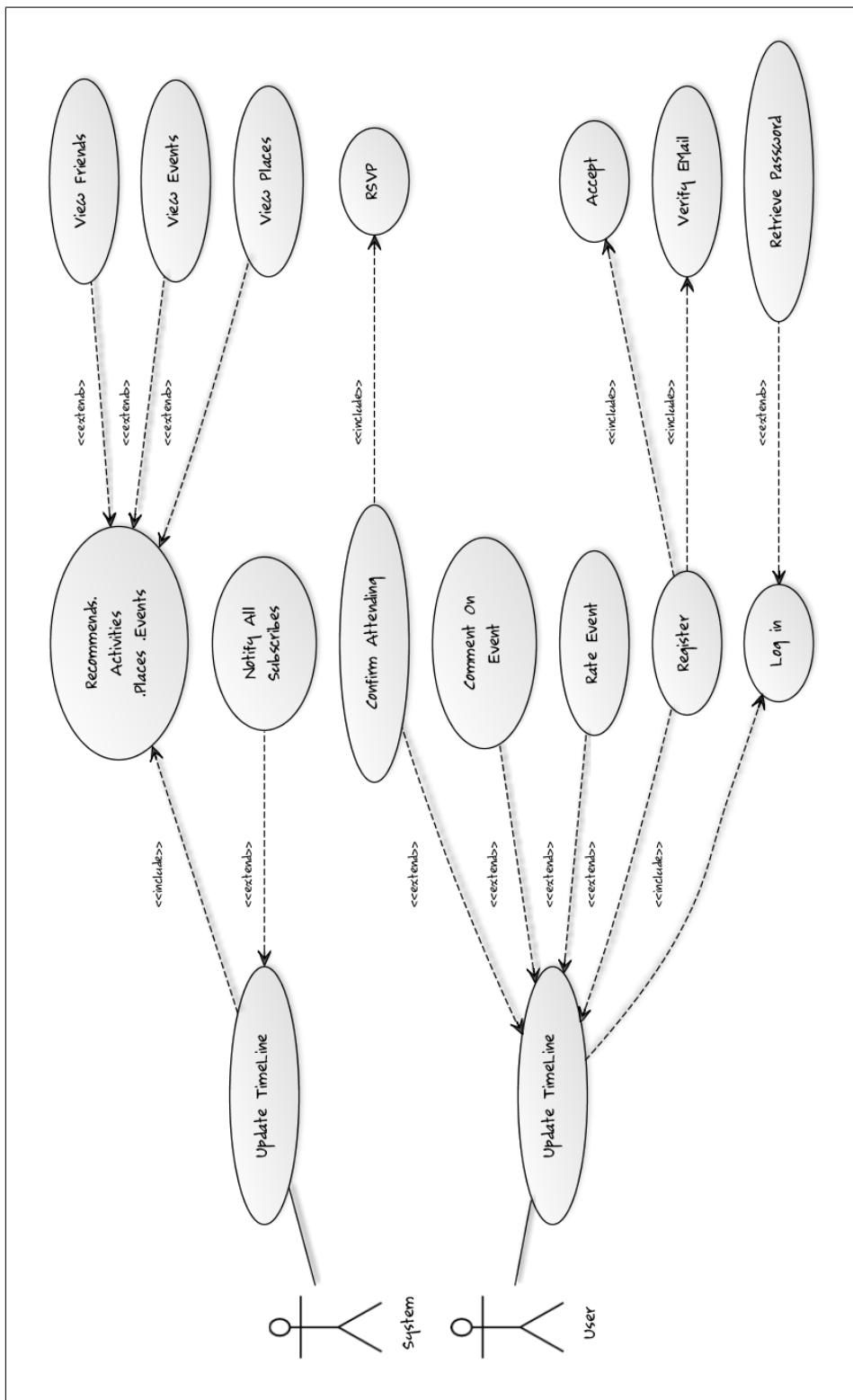


Figure 4.9: Web-Use Case-Update Timeline

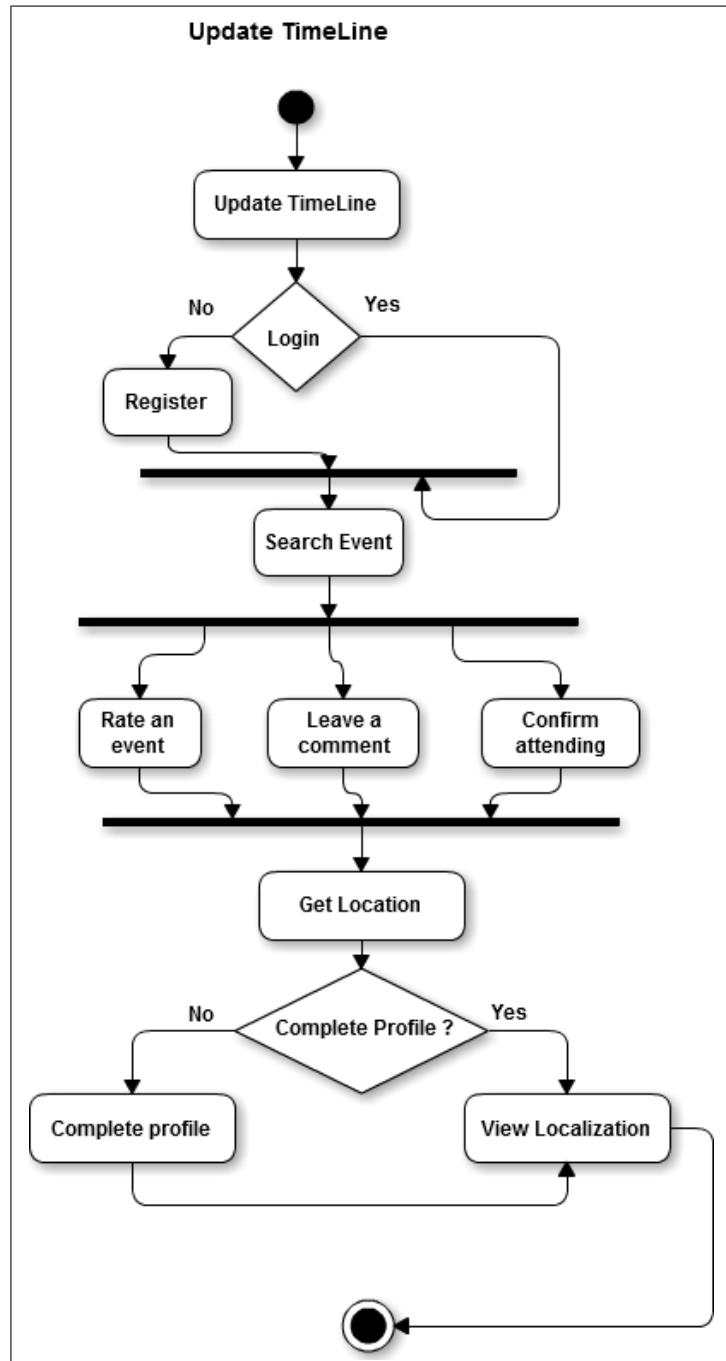


Figure 4.10: Web-Activity-Update TimeLine

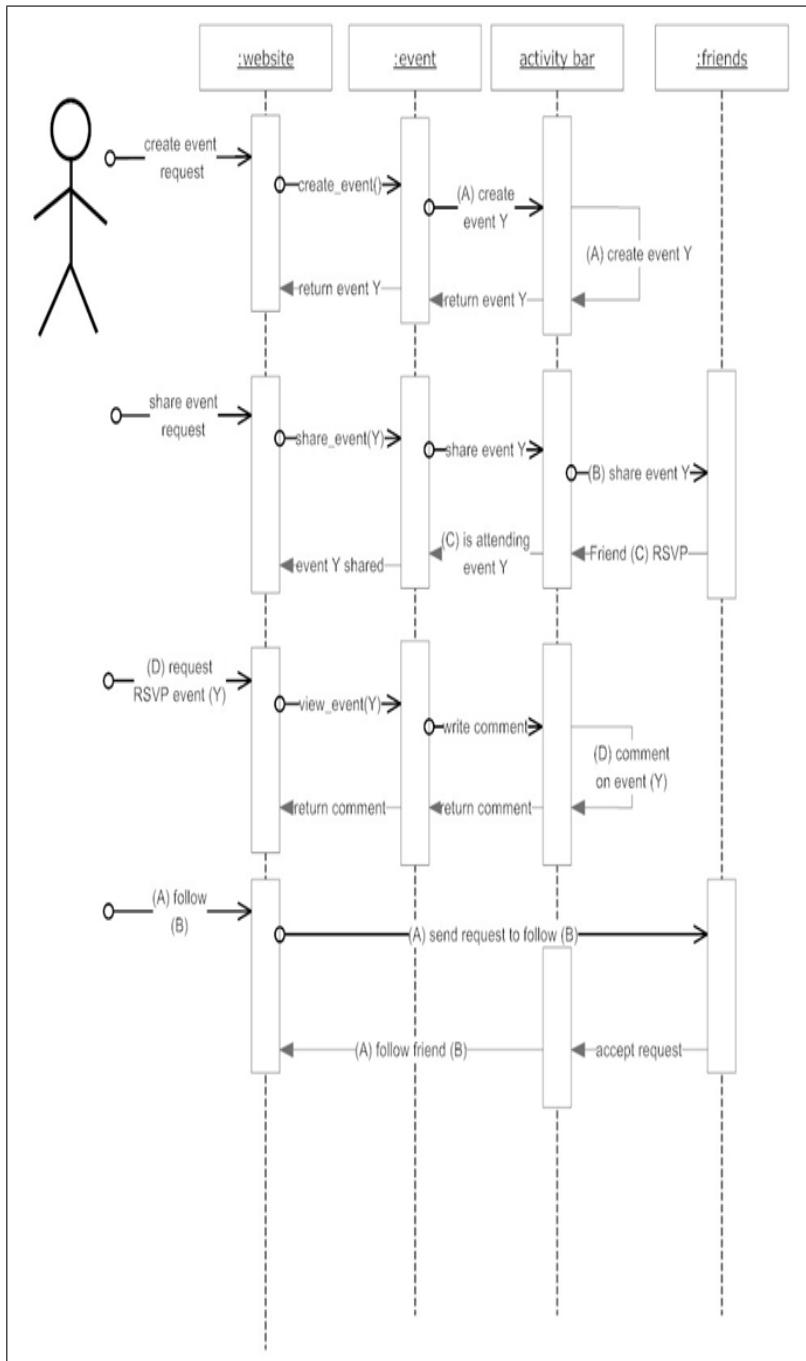


Figure 4.11: Web-Sequence-Update TimeLine

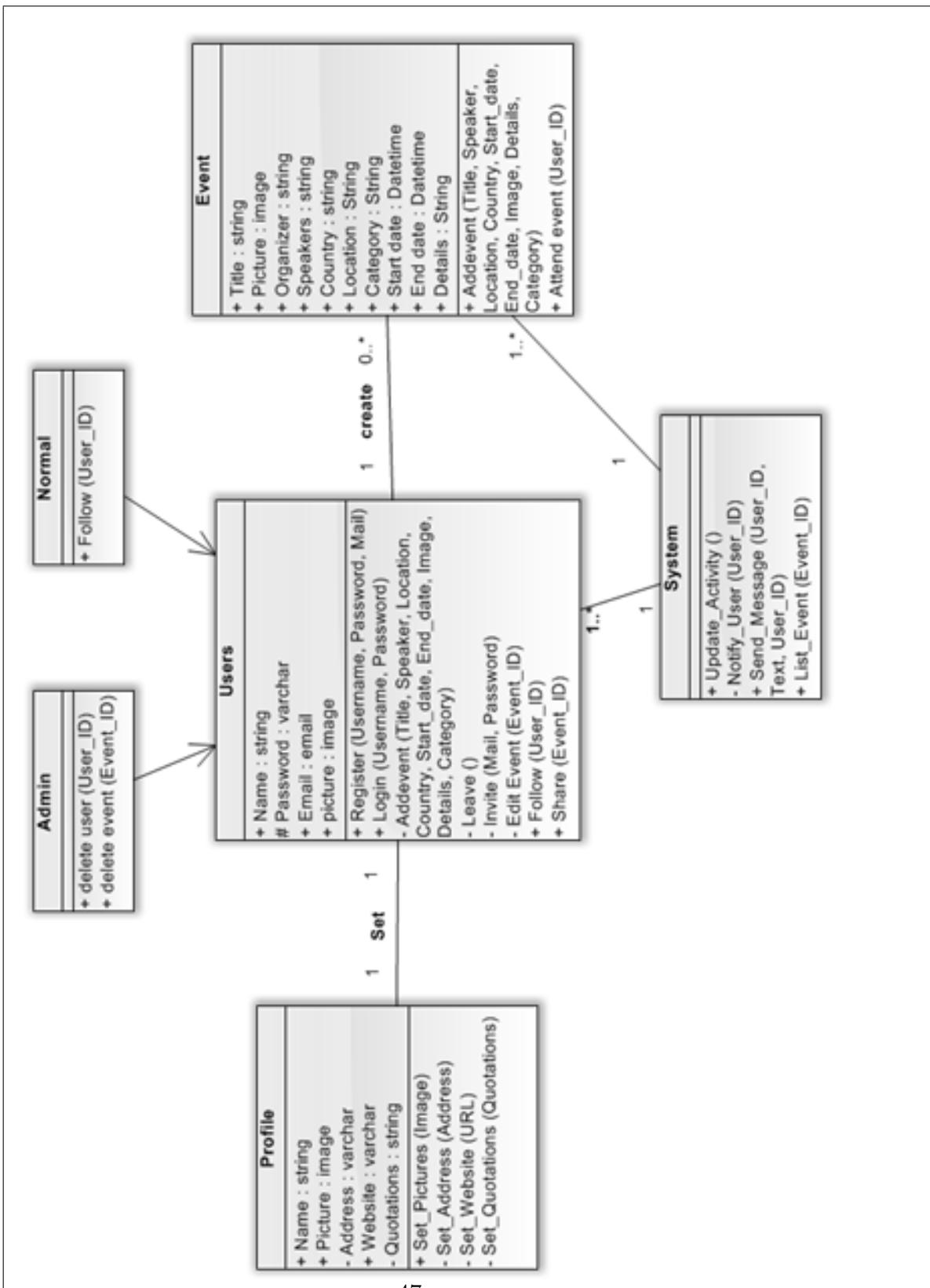
Actor	User (Authorized People)and System
Description	Users need friends to communicate with each other, so s/he can search and find friends by mail and send them a request for adding. Users can also edit his/her profile.
Data	Requests for friendship send to each other.
Stimulus	User command issued by authorized people .
Response	Users receive accepting or rejecting.
Comments	User must be authorized to make this process, he/she can get location of his/her friends.

Table 4.3: Description of Update TimeLine

4.3 website back-end design

One of the important levels of web application implementation is back-end design, it represents dealing with Database including creating, deleting, modifying, and retrieving data, in this project because of depending on Django framework for implementing the web application which is automatically integrate with SQLite database, we need first to design the database by designing Class Diagram Figure 4.12 presented in page 47 shows Class Diagram and Figure 4.13 presented in page 48 shows Entity Relationship Diagram "ERD". and Figure 4.14 presented in page 49 shows another type Entity Relationship Diagram "ERD".

Now we can begin to implement the database of the project, in Django you must first create an application by listing this command in the terminal `python manage.py startapp` application's name it will create an application in your directory then you must insert this application's name in the setting file under the category of installed application after that you can create a table for this application by listing the attributes in the application's model and this is an example of creating table for events which goes as follow in Figure 4.15 presented in page 50 shows Application Model and by a powerful command `python manage.py syncdb` it creates a table of entered attributes and automatic generate SQLite commands for this process and this is an example of creating table for events which goes as follow in Figure 4.16 presented in page 50 shows `python manage.py syncdb`. And after creating tables of database you can import it to the view of the project to apply methods of creating, deleting, and updating data in the database .



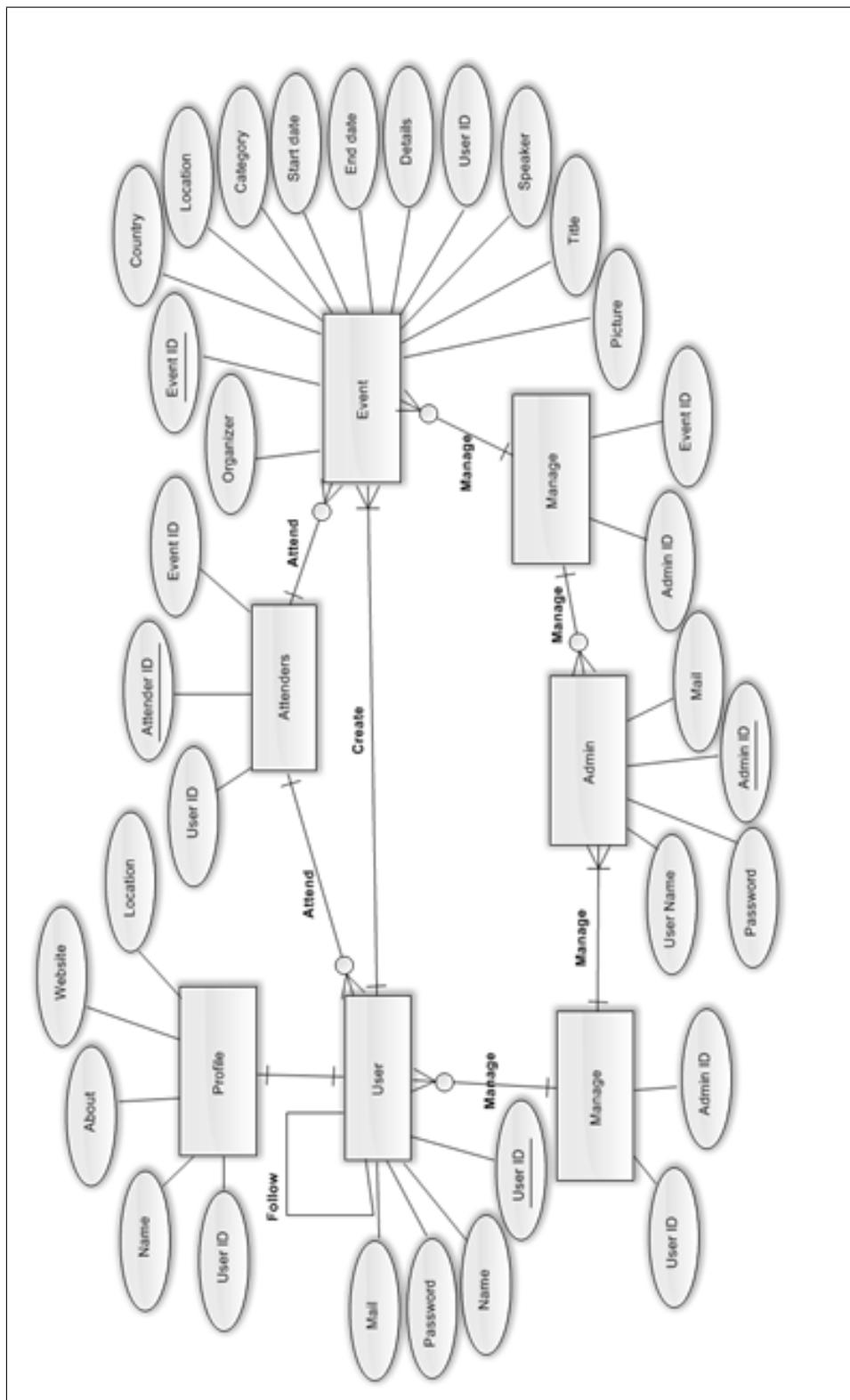


Figure 4.13: Entity Relationship Diagram
48

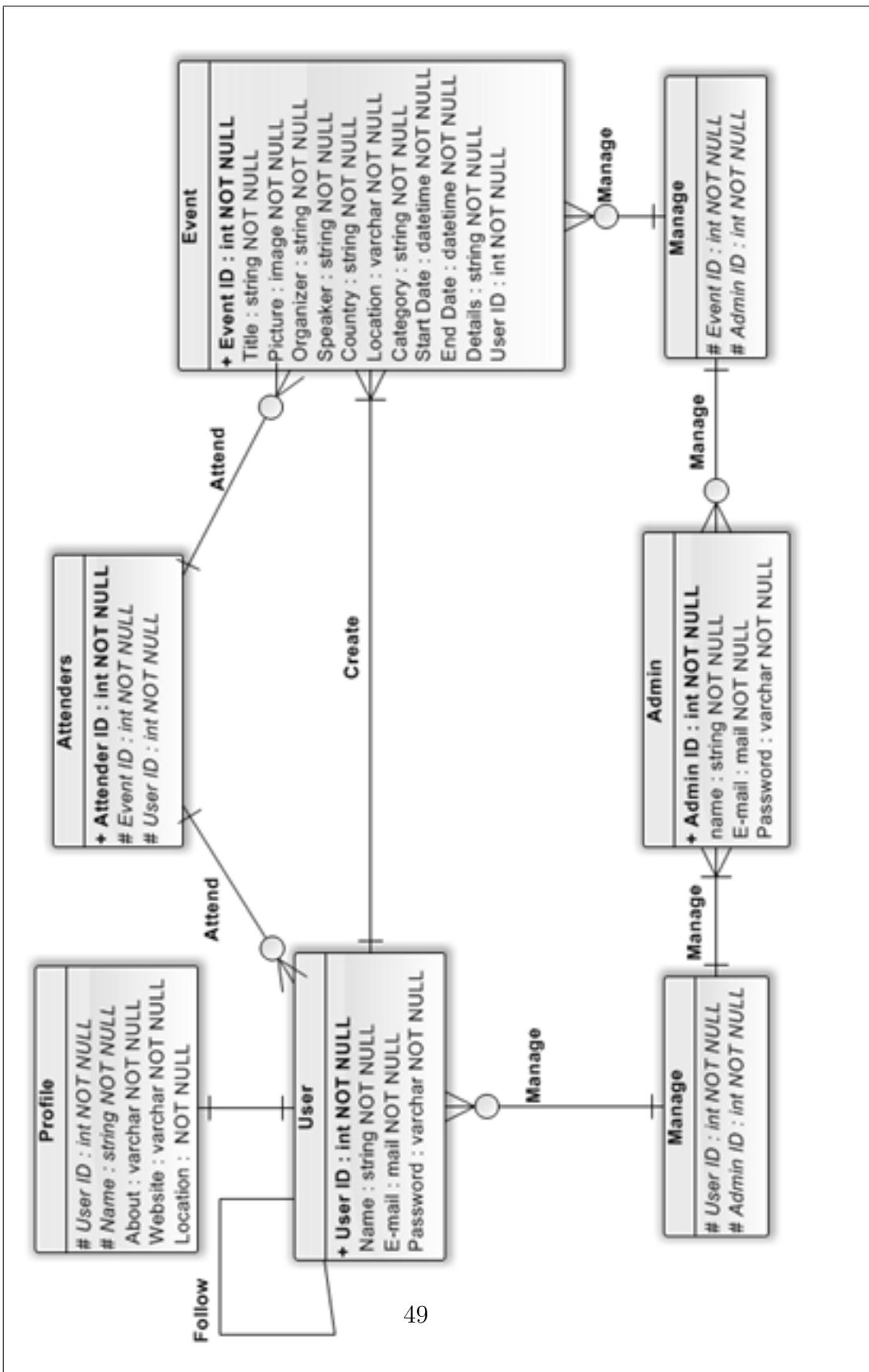


Figure 4.14: Entity Relationship Diagram 2

```

models.py
from django.db import models
# Create your models here.
from django.db import models
from django.forms import ModelForm
import datetime

class event(models.Model):
    title = models.CharField(max_length=100)
    # photo = models.ImageField(upload_to="/home/mohamed/workspace/eventati/static")
    country_choices = ((u'a',u'America'),(u'b',u'Egypt'),
                        (u'c',u'Saudi Arabia'),(u'd',u'Tunisia'),
                        (u'e',u'Lybia'),(u'f',u'Syria'))
    country = models.CharField(max_length=6 , choices = country_choices )
    location = models.CharField(max_length=100)
    category = models.CharField(max_length=50)
    organizer = models.CharField(max_length=50)
    speaker = models.CharField(max_length=50)
    start_date = models.DateTimeField()
    end_date = models.DateTimeField()
    free = models.BooleanField()
    details = models.TextField(max_length=900)

    def __unicode__(self):
        return self.title

#class eventform(ModelForm):
#    class Meta:
#        model = event

```

Figure 4.15: Application Model

```

(eventati-env)mohamed@Mohamed:~/workspace/eventati$ python manage.py sql event
BEGIN;
CREATE TABLE "event_event" (
    "id" integer NOT NULL PRIMARY KEY,
    "title" varchar(100) NOT NULL,
    "country" varchar(6) NOT NULL,
    "location" varchar(100) NOT NULL,
    "category" varchar(50) NOT NULL,
    "organizer" varchar(50) NOT NULL,
    "speaker" varchar(50) NOT NULL,
    "start_date" datetime NOT NULL,
    "end_date" datetime NOT NULL,
    "free" bool NOT NULL,
    "details" text NOT NULL
)
;
COMMIT;
(eventati-env)mohamed@Mohamed:~/workspace/eventati$ 

```

Figure 4.16: python manage.py syncdb

4.4 Interface Design

In this part of the chapter we will talk about pages we have and how they are working, how user accesses them, what is required from the user and how the system manage them and store them then representing them to the user. Here are some of that pages we have:

- Landing page.
- Home page.
- Create events page.
- User's profile page.
- User setting page.

4.4.1 Landing page

The first page that the users will see when they visit our site, in which we will find a video to explain our idea for the site and represent it to the users. In that page there are two things required from the user:

1. Asking the user to login using his e-mail and password.
2. If the user hasn't registered yet, then ask him to create his own account by filling the registration form which contains the required information (user-name, password, valid e-mail, location), then the user is allowed to access our site.

Figure 4.17 presented in page 52 Here is how the landing page looks like

4.4.2 Home page

The first page that the user will see after logging in, in which there will be the upcoming events, most recent events, the categories by which we can make finding your interested events easier and more effective. We also find the activity bar that shows you your friends activities, what they share, what they attend and their comments, so it can also be given comments and to be aware of all updates happening around you every second. Map, by which system can detect your current location and show you the events around your area. Map not only shows events around you, but also allows you to determine the location of the event you will create or attend. Here is an example for the most recent events:



Figure 4.17: Landing Page

```

1 def list_fn(request):
2     comming_soon_event_list=event.objects.all().order_by('-start_date')[:5]
3     return render_to_response('index.html',{'comming_soon_event_list':comming_soon_event_list})

```

Figure 4.18: Example Code Of last 5 Events

```

1 from django.conf.urls.defaults import *
2 from django.views.generic.simple import direct_to_template
3 from add_event.views import *
4 # Uncomment the next two lines to enable the admin:
5 from django.contrib import admin
6 admin.autodiscover()
7
8 urlpatterns = patterns('add_event.views',
9                         (r'^add_event/$', "addevent"),
10                        (r'^index/$', "list_fn"),
11
12                        # Examples:
13                        # url(r'^$', 'addevent.views.home', name='home'),
14                        # url(r'^addevent/', include('addevent.foo.urls')),
15                        # (r'^add_event/$', addevent),
16                        # Uncomment the admin/doc line below to enable admin documentation:
17                        # url(r'^admin/doc/', include('django.contrib.admindocs.urls')),
18
19                        # Uncomment the next line to enable the admin:
20                        url(r'^admin/', include(admin.site.urls)),
21
22)

```

Figure 4.19: Code Of Calling Function

Example 1: Here is the method we use to request the last 5 events recently added by users to be viewed at the top. Figure 4.18 presented in page 53 And now it's the calling function to be added the page and specify a space for it in the database Figure 4.19 presented in page 53 shows code of calling function. Figure 4.20 presented in page 54 shows The html code for the page Figure 4.21 presented in page 54 shows the final output.

4.4.3 Create Events

Create event, by which the user or the company has the ability to create his own event. In this page there are some fields required from the user:

```
1  {# if comming_soon_event_list #}
2  <h> comming_soon_ event </h1>
3  


4  {# for event in comming_soon_event_list #}
5
6  <li><a href="{{ event.title }}/">{{ event.title }}</a></li>
7  {# endfor #}
8  

9  {# else #}
10 <p>No event are available.</p>
11 {# endif #}
```

Figure 4.20: The html code of page

```
comming_soon_ event
• cairo sw week end
• GTUG meeting
• english course
• google day
• microsoft day
```

Figure 4.21: The final output

- Event name.
- Photo for the event.
- Country.
- Location.
- Set event's location on the map.
- Starting and ending date.

And many other details required about the event. We receive the entered information and then it is stored in the database to be represented to the user. We create a model by which we make the initial design or interface of the page and its contents. Then we create a method to call the information entered by the user to save them in the database and return the page to the user. Figure 4.22 presented in page 56 shows Example 1 Of Create event and Figure 4.23 presented in page 56 shows Example 2 Of Create event. The previous example represents the method that we use to call the information the user has entered , and now we will show another example that shows us the method that make the fields in which the user will write the information required from them. The Figure 4.24 presented in page 57 shows the result output.

4.4.4 Profile Page

Figure 4.25 presented in page 57 shows The user's profile page is built in the pinax and here is the html code for the page to get the values from the user: Figure 4.26 presented in page 58 Shows the page as the following.

4.4.5 Setting Page

Figure 4.27 presented in page 58 Shows The page in which user can control the account, the privacy setting and it's also a built in page in pinax.

4.5 Implementation of testing

4.5.1 Testing

Automated testing is an extremely useful bug-killing tool for the modern Web developer. You can use a collection of tests to solve, or avoid, a number of problems:

The screenshot shows a code editor window with three tabs at the top: 'homepage.html', 'site_base.html', and 'views'. The 'views' tab is active and contains the following Python code:

```
from django.template import RequestContext
from django.http import HttpResponseRedirect, HttpResponse
from django.shortcuts import render_to_response
from event.forms import *
#
def addevent(request):
    #
    if request.method == 'POST':
        form = eventform(request.POST)
        if form.is_valid():

            form.save()
        else:
            form = eventform()

    variables = RequestContext(request, {
        'form': form
    })
    return render_to_response('createevent.html', variables)
```

Figure 4.22: Code 1 Of create Event

The screenshot shows a code editor window with one tab labeled 'models'. The code defines a Django model named 'event' with the following fields:

```
from django.db import models

# Create your models here.
from django.db import models
from django.forms import ModelForm
import datetime

class event(models.Model):
    title = models.CharField(max_length=100)
    # photo = models.ImageField(upload_to="home/mohamed/workspace/eventatil/static")
    country_choices = ((u'a',u'America'),(u'b',u'Egypt'),
                       (u'c',u'Saudi Arabia'),(u'd',u'Tunisia'),
                       (u'e',u'Lybia'),(u'f',u'Syria'))
    country = models.CharField(max_length=6 , choices = country_choices )
    location = models.CharField(max_length=100)
    category = models.CharField(max_length=50)
    organizer = models.CharField(max_length=50)
    speaker = models.CharField(max_length=50)
    start_date = models.DateTimeField()
    end_date = models.DateTimeField()
    free = models.BooleanField()
    details = models.TextField(max_length=900)

    def __unicode__(self):
        return self.title

#@class eventform(ModelForm):
#    class Meta:
#        model = event
```

Figure 4.23: Code 2 Of create Event

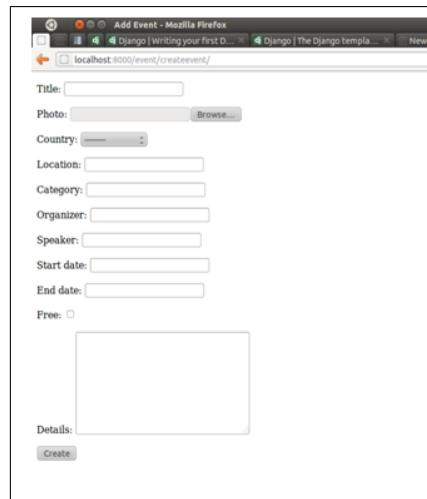


Figure 4.24: The Result Out Put

```
{% load account_tags %}


{% if profile.name %}
        <a href="{% url profile_detail profile.user.username %}">{{ profile.name }} ({{ user_display profile.user }})</a>
    {% else %}
        <a href="{% url profile_detail profile.user.username %}">{{ user_display profile.user }}</a>
    {% endif %}



{% if profile.location %}{{ profile.location }}{% endif %}
    {% if profile.location and profile.website %}&bull;{% endif %}
    {% if profile.website %}<a href="{{ profile.website }}>{{ profile.website }}</a>{% endif %}



</div>


```

Figure 4.25: Code Of user's Profile

Edit Profile

Name

About

Location

Website

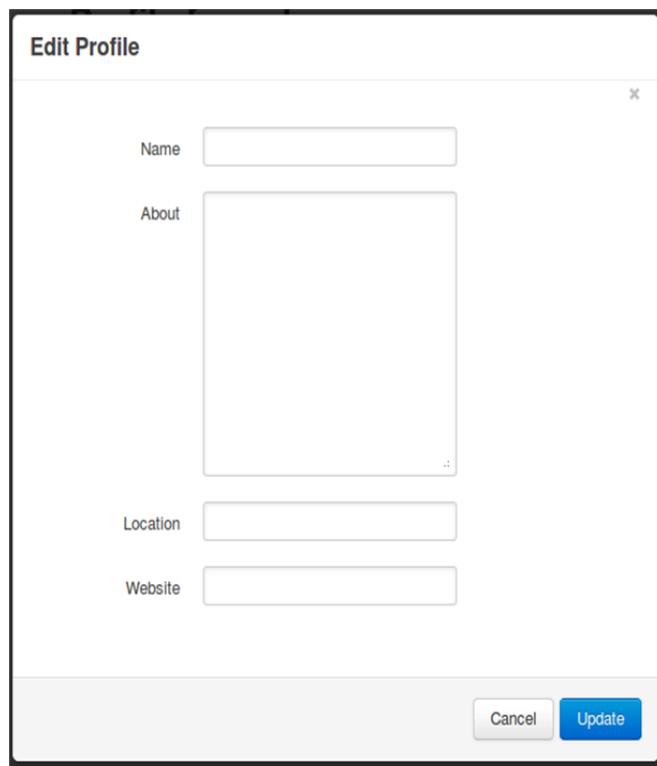


Figure 4.26: user's profile Page

Account

[Email Addresses](#)

[Change Password](#)

[Language](#)

[Timezone](#)

Email Addresses

enter@yahoo.com verified - primary

Add Email Address

Email

© 2011 <your company here>

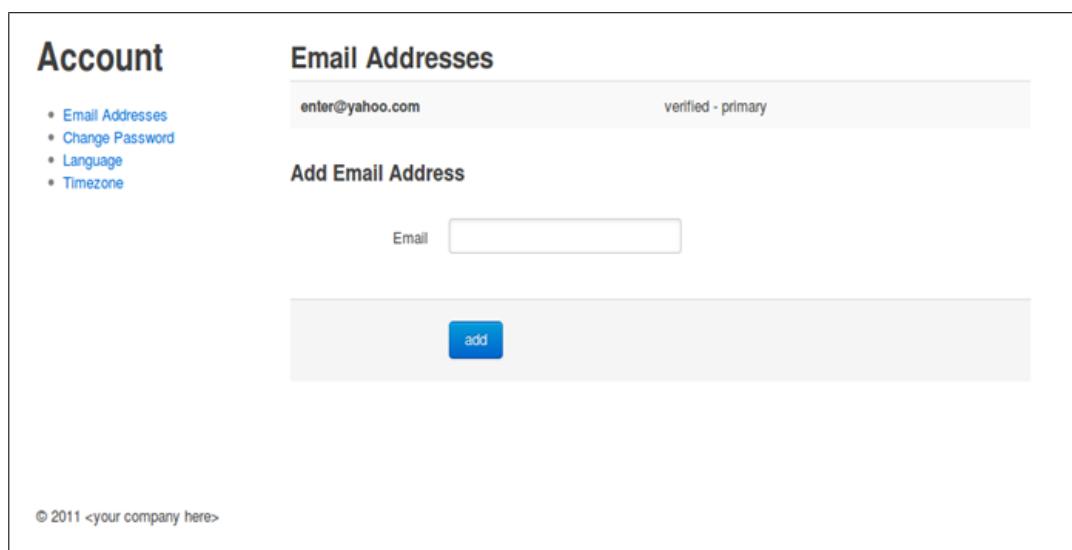


Figure 4.27: user's profile Page

- When you're writing new code, you can use tests to validate your code works as expected.
- When you're re-factoring or modifying old code, you can use tests to ensure your changes haven't affected your application's behaviour unexpectedly.

4.5.2 Why testing ?

”Code without tests is broken by design.” - Jacob. Providing automated tests for your code is a way to repeatedly ensure, with minimal developer effort, that the code you wrote to handle a task works as advertised. I like to think of tests as my insurance policy. They generally keep me from breaking existing code and looking foolish to other people. They’re also concrete proof that the code works correctly. Without that proof, what you have is a pile of code that worked right once on your machine and that you’ll either have to hand-test again and again in the future or will break without you knowing any wiser. When you first get started, writing tests is a scary task that sounds like extra work. But simple tests are easy to write and having some tests is better than no tests at all. And as you add new tests, your suite (and your confidence) grows with it. This is not to say that tests solve everything. There will always be bugs in software. Maybe the tests miss a code-path or a user will use something in an unexpected way. But tests give you better confidence and a safety net.

4.5.3 When testing?

Another point of decision is deciding whether to do **test-first** (a.k.a. Test Driven Development) or test-after. *Test-first* is where you write the necessary tests to demonstrate proper behaviour of the code **BEFORE** you write the code to solve the problem at hand. **Test-after** is when you've already written the code to solve the problem, then you go back and create tests to make sure the behaviour of the code you wrote is correct.

4.5.4 Writing testing

With Django (and Python in general) there are two main ways to write tests for your projects test suite, doctests and unit tests. Doctests are written in Python docstrings and unit test are defined with classes. firstly,

- Unit tests – tests that are expressed as methods on a Python class that

```
def add_two(num):
    "Return the result of adding two to the provided number."
    return num + 2
```

Figure 4.28: Docstrings

subclasses unittest.TestCase or Django's customized TestCase. that helps in "Eventati" and we will show that later.

For a given Django application, the test runner looks for unit tests in two places:

- The models.py file. The test runner looks for any subclass of unittest.TestCase in this module.
- A file called tests.py in the application directory – i.e., the directory that holds models.py. Again, the test runner looks for any subclass of unit-test.TestCase in this module.

Secondly,- Doctests – tests that are embedded in your functions's docstrings and are written in a way that emulates a session of the Python interactive interpreter. A docstring is a string literal that occurs as the first statement in a module, function, class, or method definition Figure 4.28 presented in page 60 Shows The Example of Docstrings As with unit tests, for a given Django application, the test runner looks for doctests in two places:

- The models.py file. You can define module-level doctests and/or a doctest for individual models.
- A file called tests.py in the application directory – i.e., the directory that holds models.py. This file is a hook for any and all doctests you want to write that aren't necessarily related to models. This example doctest is equivalent to the example given in the unit-test .

4.5.5 Handelling Testing Of Eventati

Testing a Web application is a complex task, because a Web application is made of several layers of logic – from HTTP-level request handling, to form validation and processing, to template rendering. But we ensure that we must test our code in "Eventati" so we stick to unit-test. This is because tests written in unit-test run the fastest when testing Django applications and this an example of testing events view function .And to run testing write in the shell (python manage.py test event) .

Chapter 5

Mobile Components

5.1 Introduction:

Mobile devices are widely used especially in the 20's century and still in growing. Mobile devices can be Smart phones, IOS, android, Black Berry, Symbian OS, Windows Phone OS and tablets. Mobile applications are compact software programs that perform specific tasks for the mobile user. It is much easier for the mobile users to conceptualize what this means in practice with the download application on their devices and access it. statistically ,There are 4 billion phones in use and 1.08 billion are smart phones and as we know mobile applications are applications that run on smart phones or other mobile devices . That is no surprise if we take into account that 5 billion applications were downloaded in 2010. There are some serious predictions that 21 billion applications will be downloaded by 2013. Everything could be done via mobile applications, to simplify your work and life you'll likely want to find applications that will help you keep up with social networking and stay productive while on the go. The growth in mobile technologies has meant that businesses in certain sectors are even receiving most of their web traffic from users browsing in mobile devices.

The numbers of mobile users in Egypt reach to 72.94 million in February 2011, increasing from 71.50 Million in January 2011. Number of Mobile subscribers in February 2010 was 56, 49 Million subscribers [Elwaset]. With the increase in mobile subscribers in Egypt, it becomes important to target those users. So appcelerator titanium that we used to build the application.

Actor	User (Authorized People).
Description	An Authorized user can rate, leave a comment and confirm attending with RSVP to an event, find friends, Search events, create events by web view and view all events.
Data	Information that makes time line be updated or changed .
Stimulus	User command issued by authorized people and system .
Response	Confirmation or changing that happens on-line .
Comments	System must integrate with authorized user.

Table 5.1: Description of Update TimeLine For Mobile

5.2 User Requirements:

5.2.1 Update TimeLine

User requirements of mobile application that means "Use Case Diagram". In each Project we must determine requirements of users that specify the requirements the user expects from software to be constructed in a software projects. To extract it we analysis the system and represent it in a use case Diagram showing in figure 5.1 have been drawn by using this tool "Website" <http://yuml.me/> and the described table in table 5.1 in the below. Figure 5.1 presented in page 63shows Mobile-Use Case-Update TimeLine Table 5-1 description Of Update Time Line Definitely, we have many processes in our application but we prefer to collect all of them in the main processes, since our application is simple we found only one primary process is "updating time line". To update Time line we must be logged to the application but, if we're not registered, we should register to the Application to access functions of the application. After logging, you can perform your processes, you can search and view events that will be executed or done, then you decide if you want to attend or not through RSVP, then you can invite your friends and share them with each other. If you notice the importance the events you will attend you can rate them to give a feedback, or if you have a comment you can leave it. You can't only attend events but also you can create events with its attributes, if you want to announce it and determine the place of its location through a web view. You can also send requests to adding friends to communicate and share events with each others.

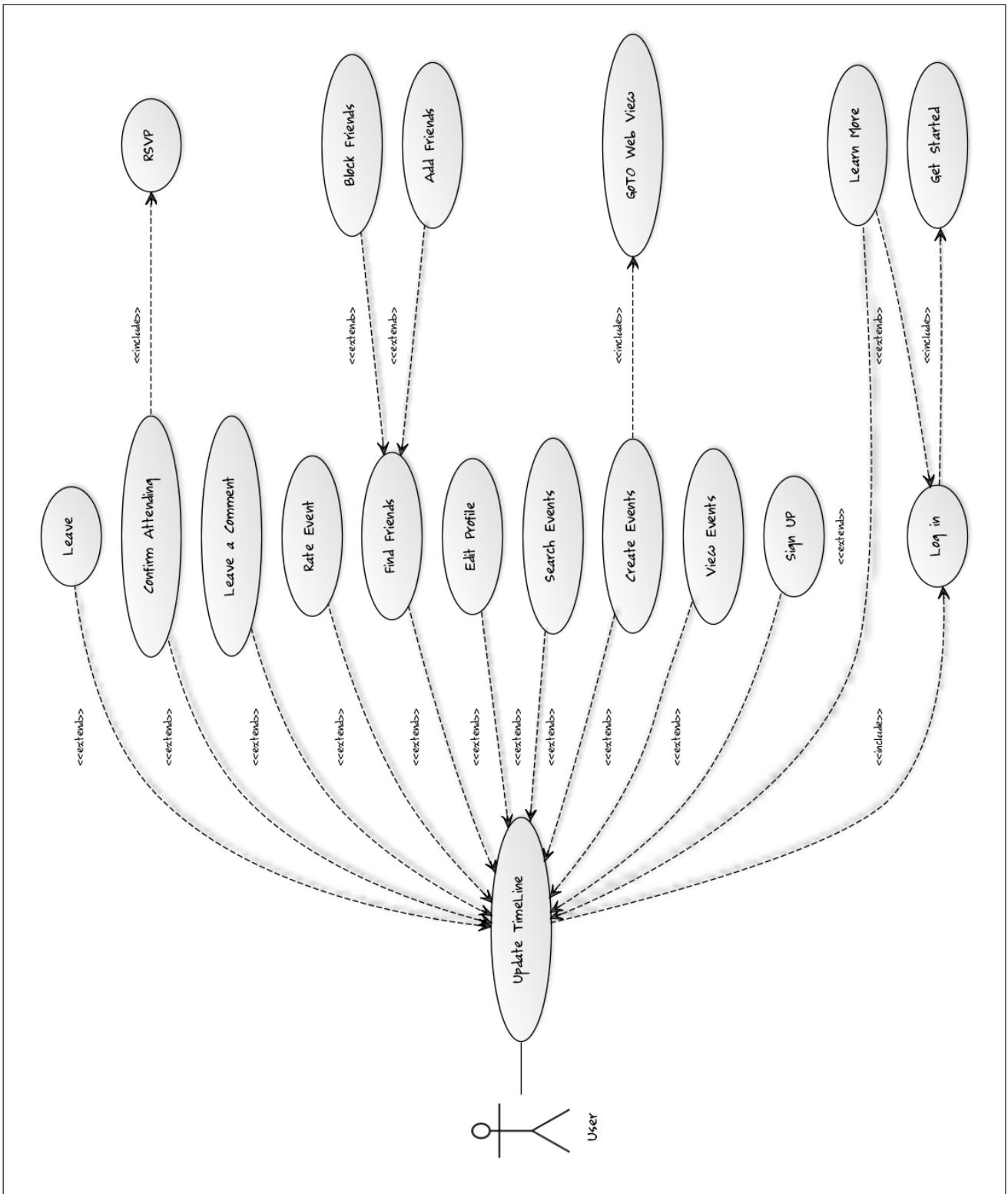


Figure 5.1: Mobile-Use Case-Update TimeLine

5.3 User Interaction with Mobile application:

User interaction with mobile applications means "Activity Diagram". In an activity diagram is a simple and intuitive illustration of what happens in a work-flow, what activities can be done in parallel, and whether there are alternative paths through the work-flow. Activity Diagram can be used to describe the operations step-by-step work-flows of components in a system. An activity diagram shows the overall flow of control. Interaction with our application is very simple not complicated at all. Figure 5.2 presented in page 65 shows Mobile-Activity-Update TimeLine. Activity Diagram in "Eventati" mobile application goes as follows:

- We first check connection, login and local file xml to start performing processes in our application, if it is yes or it is not No.
- If yes there is a connection we have another condition of login if yes login and has a local file xml, now authorized users can perform process.
- But if it isn't logged they must register to the application.
- If there is not a connection and a local xml file less than 1 day then alert that to the user, if not then install a new xml file from the web.
- After that we can search or create events through web view, then we can rate event, confirm attending, leave comments and invite Friends, then you can get location but after your complete profile.

5.4 Code of Mobile Application

"Eventati" mobile application has been developed by using Appcelerator Titanium with java script to build native mobile application and to create a version of the application for each targeted device. "Eventati" mobile application has been applied to android and I Phone OS also with the same code. Here, is a copy of an example code that we worked on it: first, Figure 5.3 presented in page 66 shows Code Of Landing page that contains logo, slogan, Get-started Function, Learn More which Contains slides of explaining Application Second: Figure 5.4 presented in page 67 shows Code Of Slides that define Application to Any Users. Third: Figure 5.5 presented in page 68 shows Code Of Login Page Which allows users entering to "Eventati" Application. fourth: Figure 5.6 presented in page 69 shows Code Of Agenda which contains a list of events will be implemented. Five: Figure 5.7 pre-

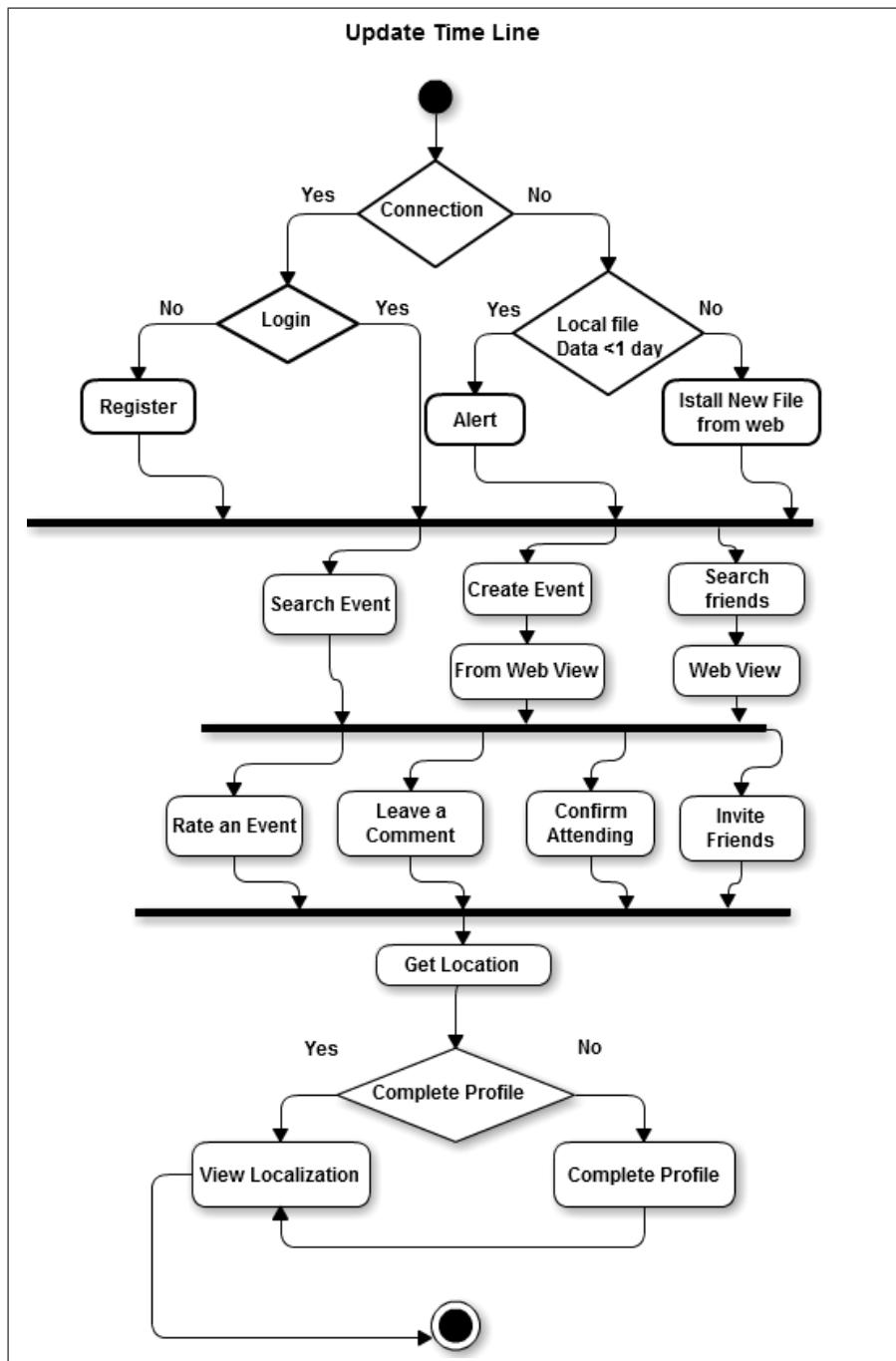


Figure 5.2: Mobile-Activity-Update TimeLine

```
1 //create the window
2 var winHome = Titanium.UI.createWindow({backgroundImage: 'icons/logo.png'});
3 winHome.orientationModes=[Ti.UI.PORTRAIT];
4 //create button
5 var btnStart = Titanium.UI.createButton({top:'50%',left:'15%',width:'105dp',height:'40dp',
6 backgroundImage:'images/getstarted.png'});
7 btnStart.addEventListener('click',
8 function(e)
9 {
10     Titanium.include('ui/login.js');
11     winHome.close();
12 }
13 );
14 winHome.add(btnStart);
15 var btnLearn = Titanium.UI.createButton({
16 top:'50%',
17 left:'60%',
18 height:'40dp',
19 width:'105dp',
20 backgroundImage:'images/learnmore.png'
21 });
22 btnLearn.addEventListener('click',
23 function(e)
24 {
25     Titanium.include('ui/img.js');
26     winLearn.show();
27 }
28 );
29 winHome.add(btnLearn);
30 winHome.open();
```

Figure 5.3: Code of Landing Page

```

1 // this sets the background color of the master UIView (when there are no windows/tab groups on it)
2 Titanium.UI.setBackgroundColor('#000');
3 //create the window
4 var winLearn = Titanium.UI.createWindow({backgroundColor:'#dddddd'});
5 winLearn.orientationModes=[Ti.UI.PORTRAIT];
6 var slider = Titanium.UI.createSlider({ top: 100, min: 0, max: 75,width: '100%', value: 0 });
7 var label = Ti.UI.createLabel({ text: slider.value, width: '100%', height: 'auto', top: 5, left: 0,
8 textAlign: Ti.UI.TEXT_ALIGNMENT_CENTER });
9 slider.addEventListener('change', function (e) { label.text = String.format("%3.1f", e.value);
10 if (slider.value==0) { view.backgroundImage='slider/1.png'; }
11 else if (slider.value==25) { view.backgroundImage='slider/2.png'; }
12 else if (slider.value==50) { view.backgroundImage='slider/3.png'; }
13 else if (slider.value==75) { view.backgroundImage='slider/4.png'; }});
14 winLearn.add(slider);
15 var view=Titanium.UI.createView({width: '100%', height: '88%', top: '0%', left: '0%', backgroundImage:'slider/1.png'});
16 // Create a Button.
17 var next = Ti.UI.createButton({height : '77dp',width : '55dp',top : '85%',left : '80%',backgroundImage:'images/forward.png'});
18 // Listen for click events.
19 next.addEventListener('click', function() { slider.value+=25;});
20 // Add to the parent view.
21 winLearn.add(next);
22 // Create a Button.
23 var back = Ti.UI.createButton({height : '77dp',width : '55dp',top : '85%',left : '0%',backgroundImage:'images/backward.png'});
24 // Listen for click events.
25 back.addEventListener('click', function() { slider.value-=25;});
26 // Add to the parent view.
27 winLearn.add(back);
28 winLearn.add(view);
29 // Create a Button.
30 var Login = Ti.UI.createButton({ height : '40dp',width : '105dp',top : '89%', left : '30%',backgroundImage:'images/getstarted.png' });
31 // Listen for click events.
32 Login.addEventListener('click', function() { Ti.include('ui/login.js');});
33 // Add to the parent view.
34 winLearn.add(Login);
35 winLearn.open();

```

Figure 5.4: Code Of the Slides Images

```

1 var login = Titanium.UI.createWindow({title:"Login", backgroundImage:"images/LoginBG.png", fullscreen:true});
2 login.orientationModes=[Ti.UI.PORTRAIT];
3 // Create a TextField.
4 var UserName = Ti.UI.createTextField({height:'40dp',top:'42%',left:'12%',width:'250dp',hintText:'User Name / Mail',color:'black',
5 softKeyboardOnFocus : Ti.UI.Android.SOFT_KEYBOARD_DEFAULT_ON_FOCUS, //Android only
6 keyboardType : Ti.UI.KEYBOARD_DEFAULT, returnKeyType : Ti.UI.RETURNKEY_DEFAULT,
7 borderStyle : Ti.UI.INPUT_BORDERSTYLE_ROUNDED});
8 //Listen for return events.
9 UserName.addEventListener('return', function(e) {UserName.blur(); alert('Input was: '+UserName.value);});
10 //Add to the parent view.
11 login.add(UserName);
12 // Create a TextField.
13 var PasswordText = Ti.UI.createTextField({height:'40dp',top:'52%',left:'12%',width:'250dp',hintText:'Password',color:'black',
14 softKeyboardOnFocus : Ti.UI.Android.SOFT_KEYBOARD_DEFAULT_ON_FOCUS, //Android only
15 keyboardType : Ti.UI.KEYBOARD_DEFAULT, returnKeyType : Ti.UI.RETURNKEY_DEFAULT,
16 borderStyle : Ti.UI.INPUT_BORDERSTYLE_ROUNDED,passwordMask:true});
17 //Listen for return events.
18 PasswordText.addEventListener('return', function(e) {PasswordText.blur(); alert('Input was: '+PasswordText.value);});
19 //Add to the parent view.
20 login.add(PasswordText);
21 // Create a Button.
22 var loginBtn = Ti.UI.createButton({backgroundImage:'images/login.png',height:'40dp',width:'250dp',top:'62%',borderRadius:1,left:'12%'});
23 var loginReq = Titanium.Network.createHTTPClient();
24 //Listen for click events.
25 loginBtn.addEventListener('click', function() {/*
26 if (UserName.value != "" && PasswordText.value != "")
27 {
28 loginReq.open("POST","http://");
29 var params = { username: UserName.value, userpass: Ti.Utils.md5HexDigest(PasswordText.value) };
30 loginReq.send(params);
31 }
32 else{
33 Ti.include('ui/agenda.js');
34 login.close();
35 alert("Username/Password are required!");
36 } });
37 //Add to the parent view.
38 login.add(loginBtn);
39 var btnLearn = Titanium.UI.createButton({top:'83%',left:'12%',height:'40dp',width:'105dp',backgroundImage:"images/learnmore.png"});
40 btnLearn.addEventListener('click',function(e)
41 {
42 Titanium.include('ui/img.js'),winLearn.show(),login.close();}),login.add(btnLearn);
43 // Create a Button.
44 var SignUp = Ti.UI.createButton({backgroundImage:"images/signup.png",height:'40dp',width:'250dp',top:'72%',borderRadius:1, left:'12%'});
45 //Listen for click events.
46 SignUp.addEventListener('click', function() { Ti.include('ui/signUp.js'); login.close();});
47 //Add to the parent view.
48 login.add(SignUp);
49 login.open();

```

Figure 5.5: Code of Login page

```

1 // create tab group
2 var tabGroup = Titanium.UI.createTabGroup();
3 var winTime = Titanium.UI.createWindow({title : 'Time Line', backgroundColor : '#dddddd'});
4 winTime.orientationModes = [Ti.UI.PORTRAIT];
5 var Timeline = Titanium.UI.createTab({title : 'Time Line',height : '15dp',window : winTime});
6 var winProfile = Titanium.UI.createWindow({title : 'Profile',backgroundColor : '#dddddd'});
7 winProfile.orientationModes = [Ti.UI.PORTRAIT];
8 var Profil = Titanium.UI.createTab({title : 'Profile',height : '15dp',window : winProfile});
9 var imgView = Titanium.UI.createView({width : '50dp',left : '2dp',height : '50dp', top : '10dp', backgroundImage : 'icons item2.png'});
10 // Create a Label.
11 var lblUserName = Ti.UI.createLabel({text : 'Eventati User', color : 'black',font : {fontSize : '16dp',fontWeight : 'bold'},height : 'auto',width : 'auto',top : '25dp', left : '100dp',textAlign : 'center'});
12 // Add to the parent
13 view.winProfile.add(lblUserName);
14 var WFI = Titanium.UI.createView({width : 'auto',left : '0dp',height : '40dp', top : '70dp', backgroundColor : '#dddddd'});
15 // Create a Button.
16 var btnFriend = Ti.UI.createButton({backgroundImage : 'images friends.png',height : '40dp', width : '105dp',top : '0dp', left : '105dp'});
17 // Create a Button.
18 var btnWall = Ti.UI.createButton({backgroundImage : 'images wall.png', height : '40dp', width : '105dp',top : '0dp', left : '0dp'});
19 // Listen for click events.
20 btnWall.addEventListener('click', function() {data.remove(searchFriend);});
21 // Add to the parent view.
22 WFI.add(btnWall);
23 var data = Titanium.UI.createView({width : 'auto', left : '0dp',height : 'auto', top : '110dp', backgroundColor : '#dddddd'});
24 var searchFriend = Titanium.UI.createSearchBar({showCancel : true, height : '40dp', top : '0dp'});
25 // print out the searchbar value whenever it changes
26 searchFriend.addEventListener('change', function(e) { //search the tableview as user types Ti.API.info('user searching for: ' + e.value);});
27 //when the return key is hit, remove focus from
28 //our searchBar
29 //when the cancel button is tapped, remove focus
30 //from our searchBar
31 searchFriend.addEventListener('cancel', function(e) {searchFriend.blur();});
32 // Listen for click events.
33 btnFriend.addEventListener('click', function() { //to our table view data.add(searchFriend);});
34 // Add to the parent view.
35 WFI.add(btnFriend);
36 // Create a Button.
37 var btnInfo = Ti.UI.createButton({backgroundImage : 'images info png',height : '40dp', width : '110dp',top : '0dp', left : '210dp'});
38 // Listen for click events.
39 btnInfo.addEventListener('click', function() { data.remove(searchFriend);
40 //alert("btnInfo was clicked!");
41 // Add to the parent view.
42 WFI.add(btnInfo);
43 winProfile.add(WFI);
44 winProfile.add(data);
45 /*
46 */
47 function clickHand(e) { if(e.rowData.nxtpage) {var win = Titanium.UI.createWindow({url : e.rowData.nxtpage,title : e.rowData.wintitle || e.rowData.title});}
48 Titanium.UI.currentTab.open(win, { animated : true });
49 /*
50 var Events = Titanium.UI.createTableView({ top : '80dp', height : 'auto',backgroundColor : '#dddddd'});
51 //to our table view
52 var searchBar = Titanium.UI.createSearchBar({showCancel : true, height : '40dp', top : '0dp'});
53 // print out the searchbar value whenever it changes
54 searchBar.addEventListener('change', function(e) { //search the tableview as user types Ti.API.info('user searching for: ' + e.value);});
55 //when the return key is hit, remove focus from
56 //our searchBar
57 searchBar.addEventListener('return', function(e) { searchBar.blur();});
58 //when the cancel button is tapped, remove focus
59 //from our searchBar
60 searchBar.addEventListener('cancel', function(e) { searchBar.blur();});
61 winTime.add(searchBar);
62 // Create a Button.
63 var CreateEvent = Ti.UI.createButton({height : '40dp', width : '105',top : '40dp', left : '70dp',backgroundImage : 'images createevent.png'});
64 // Listen for click events.
65 CreateEvent.addEventListener('click', function() { alert('you Will Goto WebSite To Create Your Event');});
66 // Add to the parent view.
67 /*
68 var file = Titanium.Filesystem.getFile("xml Eventati.xml");
69 try {
70 if(file.exists()) {var xmltext = file.read(),toString();var doc = Ti.XML.parseString(xmltext),var titleList = doc.documentElement.getElementsByName("Title");
71 //var desc=doc.documentElement.getElementsByName("title");
72 var dataArr=[];
73 for (var i = 0; i < titleList.length; i++)
74 {dataArr[i] = titleList.item(i).text;
75 var row = Titanium.UI.createTableViewRow({title : titleList.item(i).text,nxtpage:'ui\evntdetails.js',color:'black',font : {fontSize:'16dp',fontWeight:'bold', fontColor:'black'},
76 leftImage:'images events.png'});dataArr.push(row);}
77 Events.data = dataArr; Events.addEventListener('click', clickHand); } else {Ti.API.info('not found');}} catch(e) {alert(e); Ti.API.info(e);}
78 winTime.add(Events);
79 winTime.add(CreateEvent);
80 winProfile.add(imgView);
81 tabGroup.addTab(Timeline);
82 tabGroup.addTab(Profile);
83 tabGroup.open();

```

Figure 5.6: Code of Agenda

```

/**
 * @author ATY
 */
var winDetails = Titanium.UI.currentWindow;

var data=[
{title:" Event Title ", leftImage :icons/agenda1.png' ,color:'black',font:{ fontWeight:'bold', fontColor:'black'} },
{title:" Country ", leftImage :icons/location.png' ,color:'black',font:{fontWeight:'bold', fontColor:'black'} },
{title:" Location ", leftImage :icons/direction.png' ,color:'black',font:{fontWeight:'bold', fontColor:'black'} },
{title:" Category ", leftImage :icons/direction.png' ,color:'black',font:{fontWeight:'bold', fontColor:'black'} },
{title:" Organizer",leftImage :icons/organizer.png',color:'black',font:{fontWeight:'bold', fontColor:'black'} },
{title:"Speakers" ,leftImage :icons/speaker.png',color:'black',font:{fontWeight:'bold', fontColor:'black'} },
{title:" Start Date" ,leftImage :icons/calender.png',color:'black',font:{fontWeight:'bold', fontColor:'black'} },
{title:" End Date",leftImage :icons/calender.png',color:'black',font:{fontWeight:'bold', fontColor:'black'} },
{title:"Type" ,leftImage :icons/type.png',color:'black',font:{fontWeight:'bold', fontColor:'black'} },
{title:"Details" ,leftImage :icons/detail.png',color:'black',font:{fontWeight:'bold', fontColor:'black'} }
];
var listDetails=Titanium.UI.createTableView({data:data ,backgroundColor:'#dddddd'});
winDetails.add(listDetails);

```

Figure 5.7: Code Of Event details

sented in page 70shows Code Of Event details which contains attributes of each event.

Chapter 6

Design Components

6.1 Introduction

Web design means to create effective interface between technology and people which allows them to present their information and use it for meaningful purposes. The interface is a fundamental part of making the site more successful, safe, useful, functional , and in the long run more pleasurable for the user.A good website starts life in the design stage. If a website is attractive enough it can attract traffic towards it and the increase in traffic always result in the increase in business. There are several aspects of the site that are formed at this stage, including among other things, layout, color, content, functionality and maintainability.

6.2 Logo Design

The logo identifies the website via the use of a mark, flag, symbol or signature. A simple logo design allows for easy recognition and allows the logo to be versatile and memorable.In "Eventati", we used "text and symbol logo" by using "map pin" icon which portrays one of the functionalities of the website .The map pin is used to mark points on the map .similarly, "Eventati" gives the users the ability to reach the most important ,famous, popular and exciting events around the world via one click. There are many stages that we went through to reach the final result: Figure 6.1 presented in page 72 shows Every color and AscIICode we Used it. Our first try resulted in the logo in Figure 6.2 presented in page 72 in which we used the colors " f0983f" and " fafafa " and the font" Lucida Handwriting". Then, we designed the second logo in Figure 6.3 presented in page 72 in which we used the color " 69b9cb ", and the font "karabinE". Finally, we changed it as we found that

Color	Code
	#f0983f
	#fafafa
	#69b9cb
	# 9c9e9a
	# 37aabe

Figure 6.1: AscII Code Colors

it'll be difficult for the user to read it . The new logo in Figure 6.4 presented in page 72 is simple and to-the-point. The new colors sit much nicer than the old ones as We used the color "9c9e9a" to present the word event and the color "37aabe" for the entire logo and the font "MoolBoran" .

As the slogan is a very important element for a website because it makes it that much easier to increase users retention, rate and desire, we choose a slogan in Figure 6.5 presented in page 72 that represent what we are trying to help the user not to miss.



Figure 6.2: First Logo



Figure 6.3: Second Logo



Figure 6.4: Final Logo Of Eventati



Figure 6.5: Slogan Of Eventati

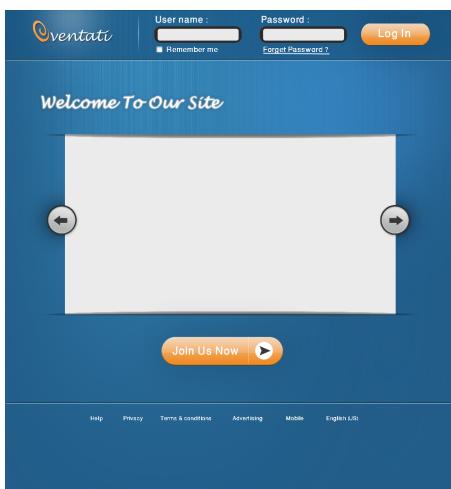


Figure 6.6: First Logo

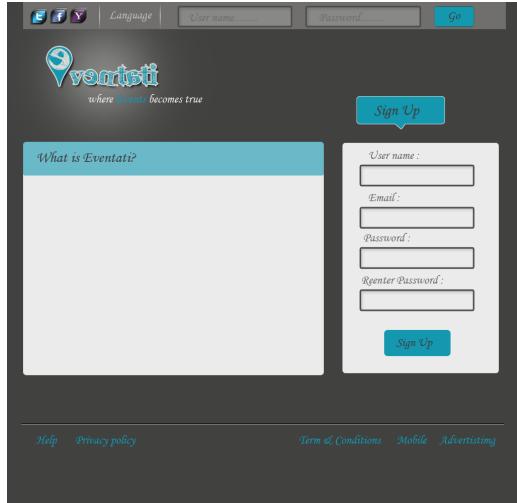


Figure 6.7: Second Logo

6.3 Web pages layouts

6.3.1 Landing page

Landing page is the first page that the user perceives upon entering the website URL at the browser addresses area.

First, we have designed page Figure 6.6 presented in page 73 show initial landing page which contain form for login process, sign up link, and slides how which contain several slides to introduce our project and our services, then we have made some changes, we have exchanged the sign up link with sign up form to facilitate the process of signing up Figure 6.7 presented in page 73 shows Second Landing Page. Finally we designed an interactive landing page which contains a login form where user-name and password can be entered. A validation is performed to verify whether the user is an already authorized user, if not the user is allowed to sign up by filling up the necessary details on the registration form. Both of login and registration forms are located in Slide Down jQuery, and exchanged the slide show with a video as Videos are one of the best methods for communicating with audiences because it's much easier to consume the visual image than to read something. The visual impact created by videos can often be creatively used to reach out and capture the imagination of a wide audience. The landing page appears as given below Figure 6.8 presented in page 74 shows The final landing page.



Figure 6.8: Final landing Page

6.3.2 Home page

The home page Figure 6.9 presented in page 76 shows the Home page of "Eventati" website is the first page that appears to the user after a successful login. The entire website depends on how the home page is designed which forms the platform for viewing other web forms. In short, the home page form is the abstract of the entire website. On home page you can review upcoming events ordered by date of creation or number of views. TimeLine exists to show current events and friends activities ordered by date when it was made, tag cloud which provide visitors with an instant illustration of the most important and popular subjects in the site dynamically.

6.3.3 Create event page

This page supports creating events, you'll be asked to input your event's name and location, as well as a start and end time, and some additional information about your events as shown in Figure 6.10 presented in page 77 shows Home Page of "Eventati".

6.3.4 Profile page

Profile page Figure 6.11 presented in page 78 is the personal community page. Showing photos, friends, some personal information, also Favourites, attended, and created events as well as allows sharing events and invite friends. Profile page can be viewed by logged in users.

6.3.5 Event Details page

After clicking on the event, you will receive event details page in Figure 6.12 presented in page 79 in which users can find all details like: event location, start and end date, event organizer, speaker, type, and the additional details about the event. As well as users find comments about the event and can post comments, also find the event latest updates. Users can share and invite guests to the event. The user who created the event can click on the event to view and edit some details about the event.

6.3.6 Search page

After typing what the users looking for, s/he will receive advanced search page Figure 6.14 presented in page 81 which contains all results about what s/he searched for. "Eventati" provides search for friends and events.

The screenshot shows the Eventati home page. At the top, there is a navigation bar with the logo 'eventati' and the tagline 'Every Moment Something Happens'. It includes a search icon, a notification icon, and a 'User Name' dropdown. Below the navigation bar is a map of the United Kingdom with major cities and roads labeled.

Below the map are two tabs: 'Recently Added' and 'Most Popular'. A large button labeled 'Create Event' with a location pin icon is positioned next to them.

The main content area features a grid of event thumbnails under the heading 'Networking Events'. Each thumbnail includes a small image of an event scene, the event name, a brief description (using placeholder text 'Lorem Ipsum'), and a 'See More' link.

To the right of the event grid is a sidebar titled 'Activities' which lists recent user attendance at networking events. Each entry shows a profile picture, the user's name ('Yara Khaled'), and the event name ('Networking Event').

At the bottom of the sidebar is a section titled 'Popular Tags' containing placeholder text about the history of Lorem Ipsum.

The footer contains links for 'Here To Help' (About Us, Contact Us, Help), 'How To' (Free Online Registration, Send Email, Publish Event Facebook Or Twitter), and social media links for Facebook, Twitter, and Google+. There are also download links for iPhone and Android, and a graphic with the text 'Wherever You Go Whenever You Need Us' featuring a blue bird icon.

Figure 6.9: Eventati Home Page

eventati
Every Moment Something Happens

User Name ▾

Create Event

Title

Photo **Browse**

Country

Start Date at

End Date at

Tags

Organiser

Speaker

Type Free Not Free

Details

Create

Here To Help

About Us
Contact Us
Help

How To

Free Online Registration
Send Email
Publish Event Facebook Or Twitter

Privacy
Site Map
Our Sponsor
Change Language

Figure 6.10: Create events Page

The screenshot shows the Eventati profile page for a user named Hayam Badr. At the top, there is a header with the Eventati logo, a search icon, a notification icon with '1', and a user dropdown menu labeled 'User Name'. Below the header, the user's profile picture is displayed, followed by their name 'Hayam Badr' and a 'Add Friend' button.

Below the profile picture, there are three tabs: 'All events', 'Created events' (which is selected), and 'Attended events'. The 'Created events' section displays a list of networking events, each with a thumbnail image of a conference hall, the event title 'Networking Events', a brief description ('Details : Lorem Ipsum is simply dummy text of the printing and typesetting industry....'), and a 'See More' link.

On the left side of the main content area, there is a sidebar titled 'friends' showing a grid of small profile pictures. A 'See More' link is located below this grid.

At the bottom of the page, there is a footer with links for 'Here To Help' (About Us, Contact Us, Help), 'How To' (Free Online Registration, Send Email, Publish Event Facebook Or Twitter), 'Privacy' (Privacy Policy, Site Map, Our Sponsor, Change Language), and social media icons for Facebook, Twitter, and Google+. There is also a section for mobile app download links (iPhone and Android) and a graphic with the text 'Wherever You Go Whenever You Need Us' featuring a blue bird icon.

Figure 6.11: Profile Page

The screenshot shows the Eventati website's event details page for a "Networking Event".

Header: The top navigation bar includes the Eventati logo ("eventati Every Moment Something Happens"), a search icon, a notifications icon (with 1 notification), and a user profile icon labeled "User Name".

Event Title: Networking Event

Section Headers: Updates, Info, Comments (active tab)

Where: Location : Diwan Bookstore Heliopolis, Al Qahirah, Egypt.

When: Start Date : Tue, 05 Jun 2012 at 08:30 am
End Date : Tue, 12 Jun 2012 at 09:30 am

Event Details:

- Organizer : Ahmed Mohamed
- Speaker : Ahmed Mohamed
- Type : Free
- Details : Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting.

RSVP:

- Attending
- Maybe Attending
- Not Attending

Add Rate : ★★★★☆

Attendees: A grid of 10 small profile pictures of attendees.

[More attendees](#)

Share Options:

- Share On twitter
- Share On facebook
- Share On google +

Footer:

- Here To Help:** About Us, Contact Us, Help
- How To:** Free Online Registration, Send Email, Publish Event Facebook Or Twitter
- Privacy:** Privacy Policy, Site Map, Our Sponsor, Change Language

Social media icons: Facebook, Twitter, Google+.

A blue bird icon with a speech bubble containing the text "Wherever You Go Whenever You Need Us".

Download links for iPhone and Android.

Figure 6.12: Event Details page

The screenshot shows the search results for the Eventati mobile application. At the top, there is a header bar with the Eventati logo, a search icon, and a user profile section. Below the header is a search bar with the word "Search". To the left of the search results is a sidebar with three main options: "All Result" (selected), "Find Friends", and "Events".

All Result:

- Networking Events**
Details : is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took.
[See More >](#)
- Networking Events**
Details : is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took.
[See More >](#)
- Networking Events**
Details : is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took.
[See More >](#)

Find Friends:

- Yasmin Magdy**
is simply dummy text of the printing and typesetting
is simply dummy text
[+ Add Friend](#)
- Hayam Badr**
is simply dummy text of the printing and typesetting
is simply dummy text
[+ Add Friend](#)
- Yasmin Abdallah**
is simply dummy text of the printing and typesetting
is simply dummy text
[+ Add Friend](#)

Events:

Figure 6.13: Search page
80



Figure 6.14: Hand Drawing Mobile Icons

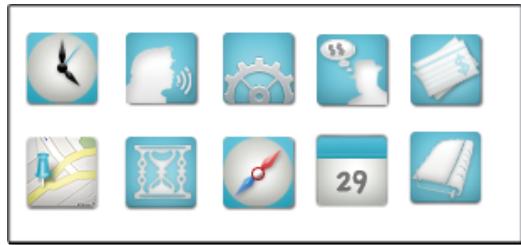


Figure 6.15: Final Mobile Icons

6.4 Design in Mobile Application

Everywhere you turn these days people are talking about mobile applications. Applications for this, applications for that. the mobile application can be used to increase the number of "Eventati" users but mobile application Can't be attractive for users without a Good Design .

6.5 Mobile Icons

For "Eventati" mobile application we designed "hand drawing icons" in size 32 and 48 pixel as shown in Figure 6.14 presented in page 81 but it didn't fit with the application so we designed another icons as shown in Figure 6.15 presented in page 81

6.5.1 Mobile layout

Figure 6.16 presented in page 82 shows Meeting that we decided the final layout which will appear when you decide to use "Eventati" mobile application. We used slide show with animated photos to introduce our application and our services, and this is the pages we have used in it. The page in Figure 6.17 presented in page 83 shows the Land Page. .Figure 6.18 presented in page 83 asks our members to login or became one of our fans by signing up . In this page Figure 6.19 presented in page 84 explains that the application gives the user the ability to create ,attend and share events. Figure 6.20 presented in page 84 shows some of our service, as our users can add friends and customize their profile settings.

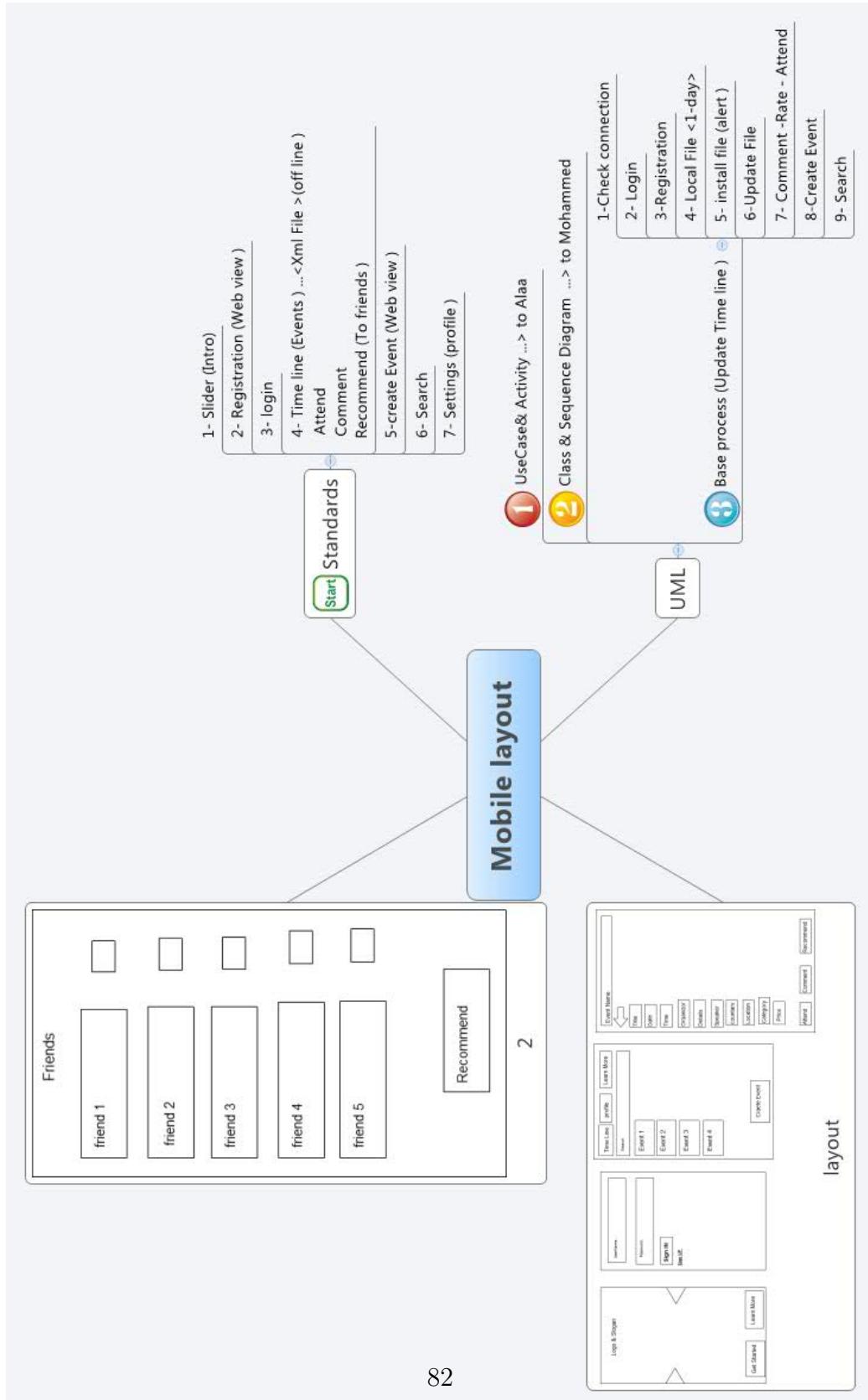


Figure 6.16: Mobile Layout

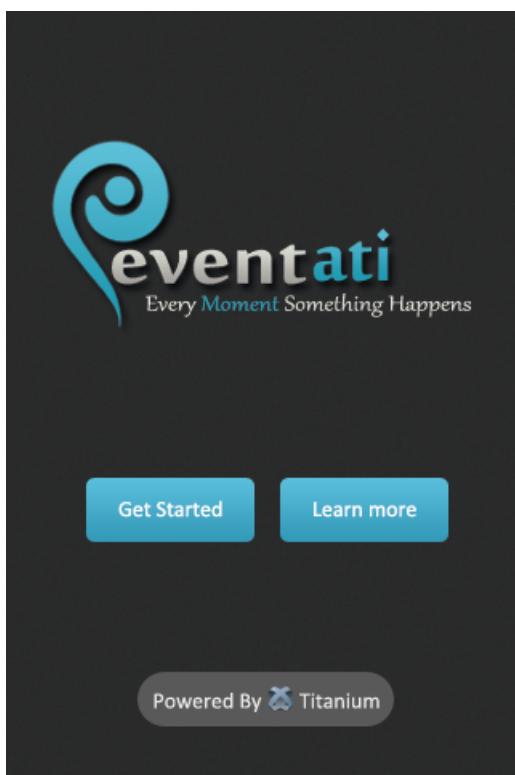


Figure 6.17: Landing Page



Figure 6.18: Slide 1



Figure 6.19: Slide 2



Figure 6.20: Slide 3

Chapter 7

Integration Between Mobile and web

7.1 Introduction

It becomes an urgent need to use mobile applications . Users increases their use for mobile phone applications and web applications will become just data warehouse.If we have a look for the (Figure 7.1) we will see how mobile use in U.S increased to become 81 minutes per day for Mobile Applications compared to 74 minutes for web consumptions. Figure 7.1 presented in page 86shows the consumption between web and mobile Application. This leads us to the importance of mobile development nowadays and also the integration between mobile and web application instead of web application.

7.2 Integration Methods

In our mobile Application we uses the most common integration method between mobile and web Applications which called Mobile Web Service . The main function of this method is to use the whole Web Application DataBase and the whole functions of DataBase (Insert , Update , Delete ,Select) to make the useful usage of mobile Application . DB Integration Method have the main advantage of data integrity which make no restrictions of misuse between Mobile user And Web User As all data warehouse is shared in the same place and all kinds of users (Mobile , Web) make the same required functions but in some difference in user interface or data required. Data required for inserting or selecting in mobile differ from web with the hope of leading best user usage of Mobile Application , In mobile application

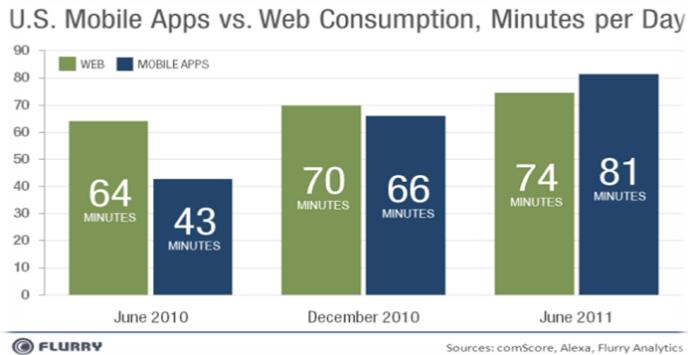


Figure 7.1: Web and Mobile consumption per day

we require less data from web . Our method to use Web Services on mobile Application in XmlRPC method.

7.3 XmlRPC

7.3.1 Overview

XML-RPC is a Remote Procedure Calling protocol that works over the Internet. An XML-RPC message is an HTTP-POST request. The body of the request is in XML. A procedure executes on the server and the value it returns is also formatted in XML. Procedure parameters can be scalars, numbers, strings, dates and can also be complex record and list structures. Figure 7.2 and 7.3 presented in page 87 and 88.

7.3.2 request Example

7.3.3 Header requirements

The format of the URI in the first line of the header is not specified. For example, it could be empty, a single slash, if the server is only handling XML-RPC calls. However, if the server is handling a mix of incoming HTTP requests, we allow the URI to help route the request to the code that handles XML-RPC requests. (In the example, the URI is /RPC2, telling the server to route the request to the "RPC2" responder.) A User-Agent and Host must be specified. The Content-Type is text/xml. The Content-Length must be specified and must be correct.

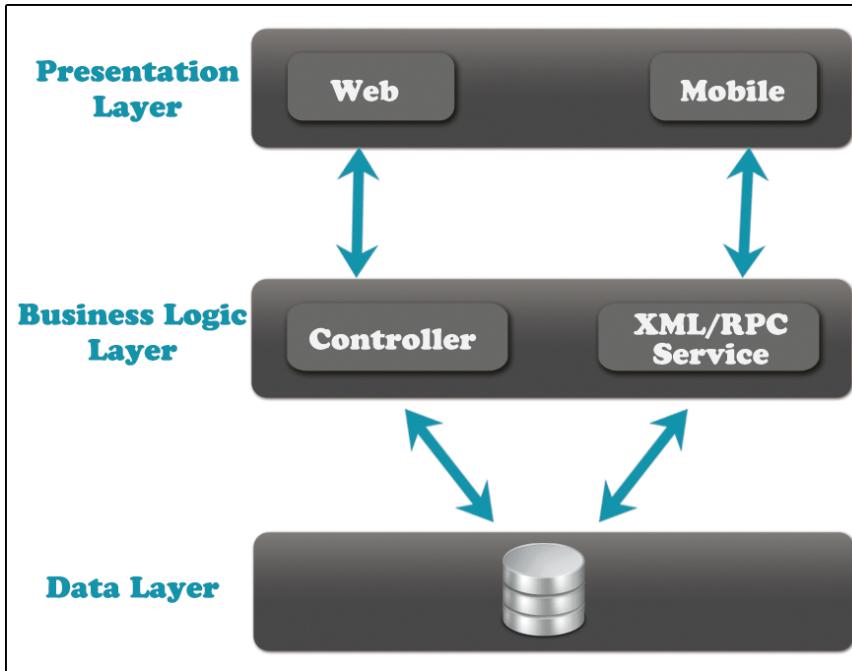


Figure 7.2: "Integration Between Mobile and Web"

7.3.4 Payload format

The payload is in XML, a single "methodCall" structure. The <methodCall> must contain a <methodName> sub-item, a string, containing the name of the method to be called. The string may only contain identifier characters, upper and lower-case A-Z, the numeric characters, 0-9, underscore, dot, colon and slash. It's entirely up to the server to decide how to interpret the characters in a <methodName>. For example, the <methodName> could be the name of a file containing a script that executes on an incoming request. It could be the name of a cell in a database table. Or it could be a path to a file contained within a hierarchy of folders and files. If the procedure call has parameters, the <methodCall> must contain a <params> sub-item. The <params> sub-item can contain any number of <params>, each of which has a <value>. Figure 7.4 presented in page 88 shows Code of Request Example .

7.3.5 Response format

Unless there's a lower-level error, always return 200 OK. The Content-Type is text/xml. Content-Length must be present and correct. The body of

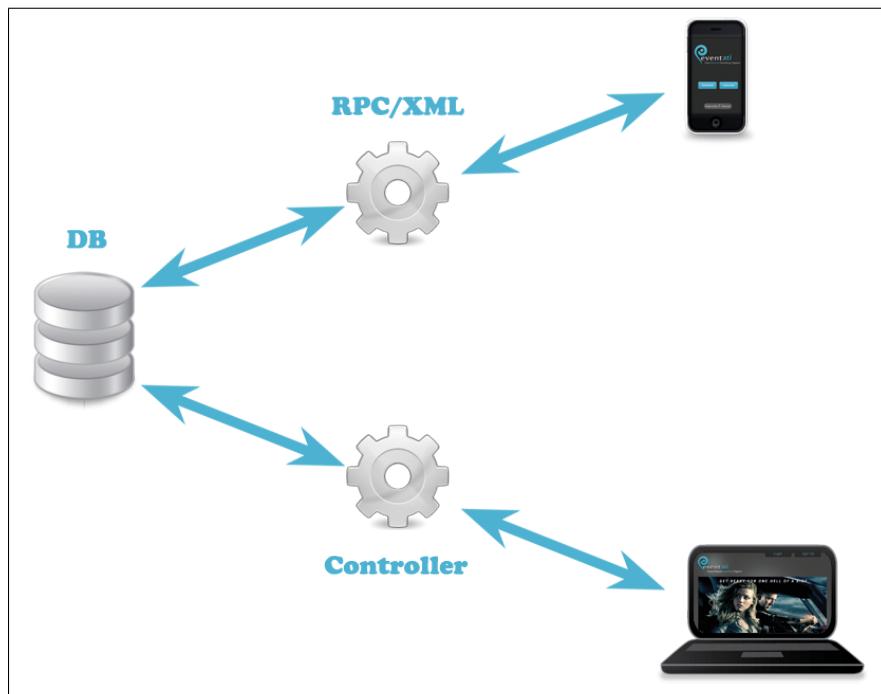


Figure 7.3: "Xml RPC"

```

POST /RPC2 HTTP/1.0
User-Agent: Eventati/5.1.2 (WinNT)
Host: eventati.com
Content-Type: text/xml
<?xml version="1.0"?>
<methodCall>
  <loginRequest>eventati.getStateName</loginRequest>
  <params>
    <param>
      <userName>mGamal</userName>
    </param>
    <param>
      <Password>123456</Password>
    </param>
  </params>
</methodCall>

```

Figure 7.4: "Request Example Code "

```

HTTP/1.1 200 OK
Connection: close
Content-Length: 158
Content-Type: text/xml
<?xml version="1.0"?>
<loginResponse>
  <params>
    <param>
      <value><string>Authorized User</string></value>
    </param>
  </params>
</loginResponse>

```

Figure 7.5: "Response Code Example "

the response is a single XML structure, a <methodResponse>, which can contain a single <params> which contains a single <param> which contains a single <value>. The <methodResponse> could also contain a <fault> which contains a <value> which is a <struct> containing two elements, one named <faultCode>, an <int> and one named <faultString>, a <string>. A <methodResponse> can not contain both a <fault> and a <params>.

7.3.6 Response Example

Here's an example of a response to an XML-RPC request. Figure 7.5 presented in page 89 shows Response Code Example "Eventati" .

7.4 Code Examples:

7.4.1 Example from Xml File:

Here is code examples from our work in "Eventati" Mobile Application. Figure 7.6 presented in page 90 shows Code Example of "Eventati" .

7.4.2 Example From UI Application Files

Example From UI Application Files of reading data from local Xml File After Downloading It from the website : Figure 7.7 presented in page 90 shows UI Application File . Figure 7.8 presented in page 91 shows UI

```

<event>
<ID>2</ID>
<Title> Conference of Arab Doctors </Title>
<Photo></Photo>
<Country> Aldakhlia</Country>
<Location> Mansoura </Location>
<Category> Health </Category>
<Organizers> Medical Association </Organizers>
<Speakers> Prof.Dr Hamdy Elsayed</Speakers>
<StartDate> Saturday,23-06-2012 </StartDate>
<EndDate> Sunday,24-06-2012 </EndDate>
<Type> Free </Type>
<Details></Details>
</event>

```

Figure 7.6: "Code Example "

```

219 var file = Titanium.Filesystem.getFile("xml/Eventati.xml");
220 try {
221 if ( file.exists() ) {
222
223 var xmltext = file.read().toString();
224 var doc = Ti.XML.parseString(xmltext);
225 var titleList = doc.documentElement.getElementsByTagName("Title");
226 var desc=doc.documentElement.getElementsByTagName("ID");
227 var Country=doc.documentElement.getElementsByTagName("Country");
228 var Location=doc.documentElement.getElementsByTagName("Location");
229 var Category=doc.documentElement.getElementsByTagName("Category");
230 var Organizers=doc.documentElement.getElementsByTagName("Organizers");
231 var Speakers=doc.documentElement.getElementsByTagName("Speakers");
232 var StartDate=doc.documentElement.getElementsByTagName("StartDate");
233 var EndDate=doc.documentElement.getElementsByTagName("EndDate");
234 var Type=doc.documentElement.getElementsByTagName("Type");
235 var Details2=doc.documentElement.getElementsByTagName("Details");
236
237 var dataArr=[];
238 var ID=[];
239 var MyCountry=[];
240 var MyLocation=[];
241 var MyCategory=[];
242 var MyOrganizers=[];
243 var MySpeakers=[];
244 var MyStartDate=[];
245 var MyEndDate=[];
246 var MyType=[];
247 var MyDetails2=[];

```

Figure 7.7: "UI Application File "

```

for (var i = 0; i < titleList.length; i++) {
    var row = Titanium.UI.createTableViewRow({
        title : titleList.item(i).text,
        nxtpage:'ui/evntdetails.js',
        color:'black',
        font:{
            fontSize:'16dp',
            fontWeight:'bold',
            color:'black'},
        leftImage:'images/events.png',
        EventID:desc.item(i).text,
        Country:Country.item(i).text,
        Location:Location.item(i).text,
        Category:Category.item(i).text,
        Organizers:Organizers.item(i).text,
        Speakers:Speakers.item(i).text,
        StartDate:StartDate.item(i).text,
        EndDate:EndDate.item(i).text,
        Type:Type.item(i).text,
    });
    ID.push(row.EventID);
    MyCountry.push(row.Country);
    MyLocation.push(row.Location);
    MyCategory.push(row.Category);
    MyOrganizers.push(row.Organizers);
    MySpeakers.push(row.Speakers);
    MyStartDate.push(row.StartDate);
    MyEndDate.push(row.EndDate);
    MyType.push(row.Type);
    dataArr.push(row);
}

```

Figure 7.8: "UI Application File2 "

Application File2 . Click Hand Function : Is A function occurs when user click on an event to show the event Details : Figure 7.8 presented in page 92 shows Event Details Code .

```
/*
-----
```

```
function clickHand(e) {
    if (e.rowData.nxtpage) {
        var win = Titanium.UI.createWindow({
            url : e.rowData.nxtpage,
            title : e.rowData.wintitle || e.rowData.title,
            ID:e.rowData.EventID,
            Country:e.rowData.Country,
            Location:e.rowData.Location,
            Category:e.rowData.Category,
            Organizers:e.rowData.Organizers,
            Speakers:e.rowData.Speakers,
            StartDate:e.rowData.StartDate,
            EndDate:e.rowData.EndDate,
            Type:e.rowData.Type,
            //Details2:rowData.Details2
        });
        Titanium.UI.currentTab.open(win, {
            animated : true
        });
    }
}
```

Figure 7.9: Event Details Code

Chapter 8

Outcomes

8.1 Web Outcome

- Figure 8.1 presented in page 96 shows The Land page of "Eventati" the numbers on this figure shows:
 1. Number 1:Shows the Video which explains and presents functions and what "Eventati" will present to users.
 2. Number 2 : Shows Login which we can login to "Eventati" by Authorized Users and passwords
 3. Number 3:Shows SignUp which any one can register to "Eventati" by UserName ,e-mail and Password.
 4. Number 4 :Shows Logo and Slogan of "Eventati" and Finally at the end of each page contains footer which contains fast links to reach the process.
- Figure 8.2 presented in page 97 shows the land page after Pressing Login Button.
- Figure 8.3 presented in page 98 shows the land page after Pressing Registration Button.
- Figure 8.4 presented in page 99 shows The Home Page of "Eventati" the numbers on this figure shows:
 1. Number 1 : Shows the Map which contains bins and location of each event.
 2. Number 2 : Shows the button which can click and create event.

- 3. Number 3 : shows events both Most popular and which recently add.
- 4. Number 4 : Shows Activities or Time Line updating that shows Authorized users activities like (Confirm Attending,Rate Event,Publish Event,leave comments on Events).
- 5. Number 5 : Shows popular tags which contains all categories of events that we interested in .
- Figure 8.5 presented in page 100 shows how to create events and their attributes .
- Figure 8.6 presented in page 101 shows The Profile Page of each user, the numbers in this figure shows:
 - 1. Number 1 : The profile Page.
 - 2. Number 2 : Shows information about authorized user.
 - 3. Number 3 : Shows user's friends and button which can use to invite friends to any events.
 - 4. Number 4 : Shows all events which user has attended it and that s/he has created them.
 - 5. Number 5 : Shows events that was created by authorized users .
 - 6. Number 6 : Shows events that was attended by authorized users.
- Figure 8.7 presented in page 102 shows Event's Details and numbers in this figure shows:
 - 1. Number 1 : Details of each event.
 - 2. Number 2 : Shows information of each event
 - 3. number 3 : Shows any updating been done on event.
 - 4. Number 4 : Shows comments been left by authorized users on event.
 - 5. Number 5 : Shows functions that authorized users can publish/share events through Social Networks like (FaceBook,Twitter,Google +) and can also invite friends to events.
 - 6. Number 6 : Shows RSVP that enable users to Confirm attending, may be or reject attending and shows Rating importance of each event.
 - 7. Number 7 : Shows Attenders of each event.

- 8. Number 8 : Shows social Networks which can share events through them.
- Figure 8.8 presented in page 103 shows search page both Events or users and can filter it according user's requests.

8.2 Mobile Outcome

In this section we will show outcomes of Mobile Application "Eventati".

- Figure 8.9 presented in page 104 shows the land page of Mobile Application , numbers in this figures show:
 1. Number 1: Get Started Button which enable you to enter application through Login process and begin using it and it's functions.
 2. Number 2: Learn More button which will present Slides to show each user the functions of the application and how to use it.
- Figure 8.10 presented in page 104 shows the Login Page,which each authorized user enter his user-name or E-mail and password.Learn More Button can each user can revise if he wants to know any thing about the application.
- Figure 8.11 presented in page 105 shows the registration Page.
- Figure 8.12 presented in page 105 shows the Home Page of the application which contains two taps shown in the numbers below.
 1. Number 1 : Shows First tap is the time Line which contains all events.
 2. Number 2 : Shows Second tap is the profile of each user.
 3. Number 3 : Shows search tap which can search on events.
 4. Number 4 : Shows function of create any event the authorized user wants but,through Web view not through the application.
- Figure 8.13 presented in page 106 shows the Details of each event.
- Figure 8.14 presented in page 106, Figure 8.15 presented in page 107 ,Figure 8.16 presented in page 107,Figure 8.17 presented in page 108 and Figure 8.18 presented in page 108 show slides of presentation that explain the application.



Figure 8.1: Land Page



Figure 8.2: Land Page After pressing login Button



Figure 8.3: Land Page After pressing Registration Button

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

918

919

920

921

922

923

924

925

926

927

928

929</

The screenshot shows the 'Create Event' page on the Eventati website. The header features the Eventati logo with the tagline 'Every Moment Something Happens'. The top right includes a search icon, a help icon, and a user account dropdown labeled 'User Name'. The main form has a dark header bar with the text 'Create Event'.

Create Event

Title: [Text input field]

Photo: [Text input field] **Browse**

Country: [Text input field with dropdown arrow]

Start Date: [Text input field] at [Text input field]

End Date: [Text input field] at [Text input field]

Tags: [Text input field]

Organiser: [Text input field]

Speaker: [Text input field]

Type: Free Not Free

Details: [Text area]

Create

Here To Help

- [About Us](#)
- [Contact Us](#)
- [Help](#)

How To

- [Free Online Registration](#)
- [Send Email](#)
- [Publish Event Facebook Or Twitter](#)

Privacy

- [Site Map](#)
- [Our Sponsor](#)
- [Change Language](#)





Figure 8.5: Create Event

The screenshot shows the Eventati profile page for a user named Hayam Badr. The interface is dark-themed with light-colored text and icons.

- Top Bar:** Includes the Eventati logo, a search icon, a notification icon with a '1' (circled as 1), and a 'User Name' dropdown.
- User Profile Section:**
 - Profile picture of Hayam Badr (circled as 2).
 - Information: studied at FCIS, from Mansoura, born on february 2, female.
 - Buttons: Invite Friends (circled as 3) and Add Friend (circled as 5).
- Navigation Tabs:** All events (circled as 4), Created events (selected tab, circled as 5), Attended events (circled as 6).
- Content Sections:**
 - Networking Events:** Details: Lorem Ipsum is simply dummy text of the printing and typesetting industry.... See More >
 - Networking Events:** Details: Lorem Ipsum is simply dummy text of the printing and typesetting industry.... See More >
 - Networking Events:** Details: Lorem Ipsum is simply dummy text of the printing and typesetting industry.... See More >
 - Networking Events:** Details: Lorem Ipsum is simply dummy text of the printing and typesetting industry.... See More >
 - Networking Events:** Details: Lorem Ipsum is simply dummy text of the printing and typesetting industry.... See More >
- Bottom Navigation:** Here To Help, How To, Privacy, Site Map, Our Sponsor, Change Language.
- Mobile Downloads:** iPhone Download, Android Download.
- Slogan:** Wherever You Go Whenever You Need Us.

Figure 8.6: Profile Page

1 User Name

2 Networking Event

3 Updates

4 Info

5 Comments

6 Where?

7 Location: Diwan Bookstore Heliopolis, Al Qahirah, Egypt

8 Start Date: Tue, 05 Jun 2012 at 08:30 am
End Date: Tue, 12 Jun 2012 at 09:30 am

9 Event Details

10 Organizer: Ahmed Mohamed
Speaker: Ahmed Mohamed
Type: Free

11 Details: Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting.

12 Share This Event

13 Invite Guests

14 RSVP

15 Attending
16 Maybe Attending
17 Not Attending

18 Add Rate:
★ ★ ★ ★ ☆

19 Attendees

20 Share On twitter

21 Share On facebook

22 Share On google +

23 Here To Help

- About Us
- Contact Us
- Help

24 How To

- Free Online Registration
- Send Email
- Publish Event Facebook Or Twitter

25 Privacy

26 Site Map

27 Our Sponsor

28 Change Language

29 iPhone Download

30 Android Download

31 Wherever You Go
Whenever You Need Us

Figure 8.7: Event Details

The screenshot shows the search results for the term "Networking Events" on the Eventati app. A large circular icon with the number "1" is positioned at the top center of the search bar. The search results are displayed in a grid format:

- Networking Events**: Three identical items are listed, each showing a thumbnail of a conference room, the title "Networking Events", a detailed description about Lorem Ipsum, and a "See More" link.
- Yasmin Magdy**: Shows a profile picture of a woman, the name "Yasmin Magdy", a short bio, and a blue "Add Friend" button.
- Hayam Badr**: Shows a profile picture of a woman with red hair, the name "Hayam Badr", a short bio, and a blue "Add Friend" button.
- Yasmin Abdallah**: Shows a profile picture of a cartoon rabbit, the name "Yasmin Abdallah", a short bio, and a blue "Add Friend" button.

At the bottom of the screen, there are links for "Here To Help", "How To", "Privacy", social media icons for Facebook, Twitter, and Google+, download links for iPhone and Android, and a graphic with the text "Wherever You Go Whenever You Need Us".

Figure 8.8: Search

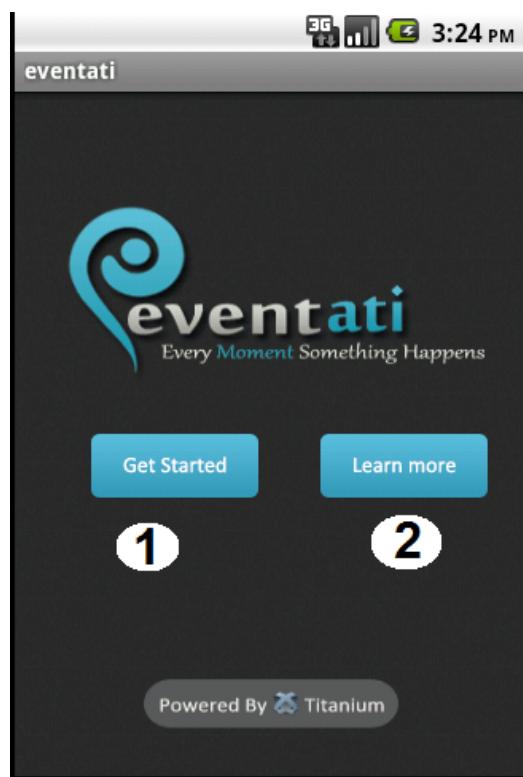


Figure 8.9: Landing Page Of Mobile

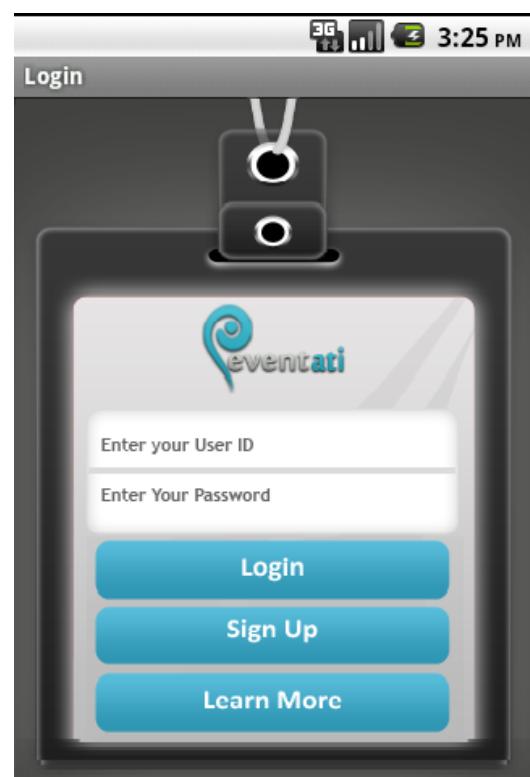


Figure 8.10: Login Page Of Mobile

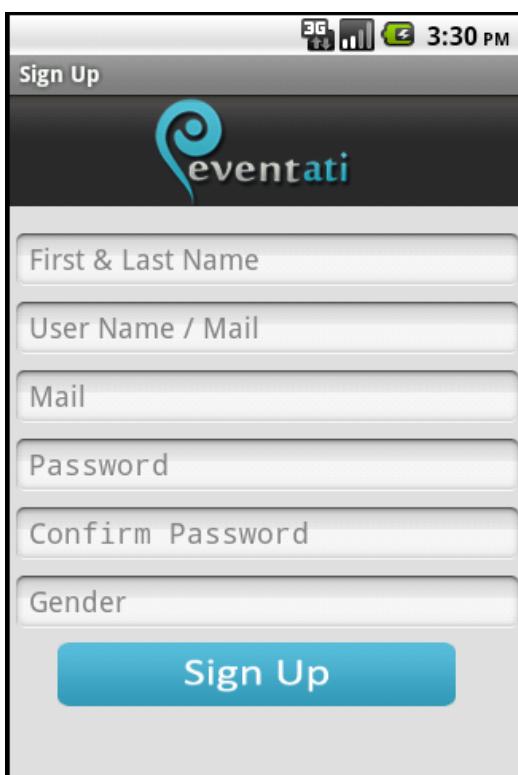


Figure 8.11: Sign-Up Page Of Mobile

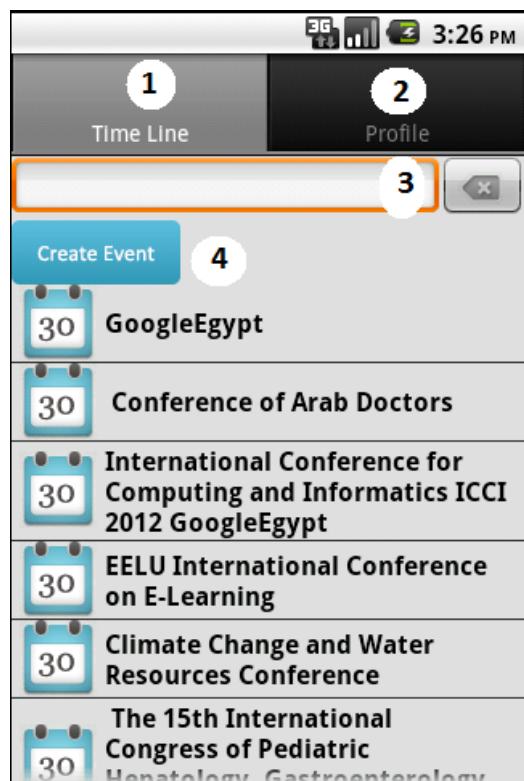


Figure 8.12: Home Page Of Mobile



Figure 8.13: Event's Details

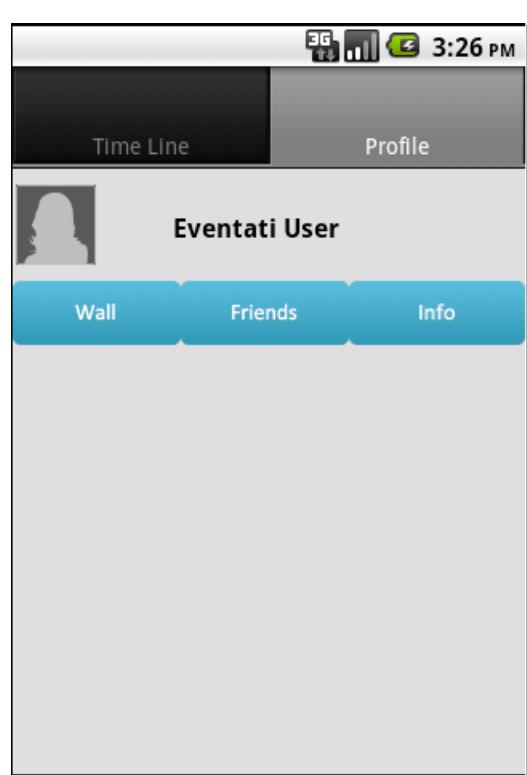


Figure 8.14: Profile Page Of Mobile



Figure 8.15: "Slide 1 Of Mobile"

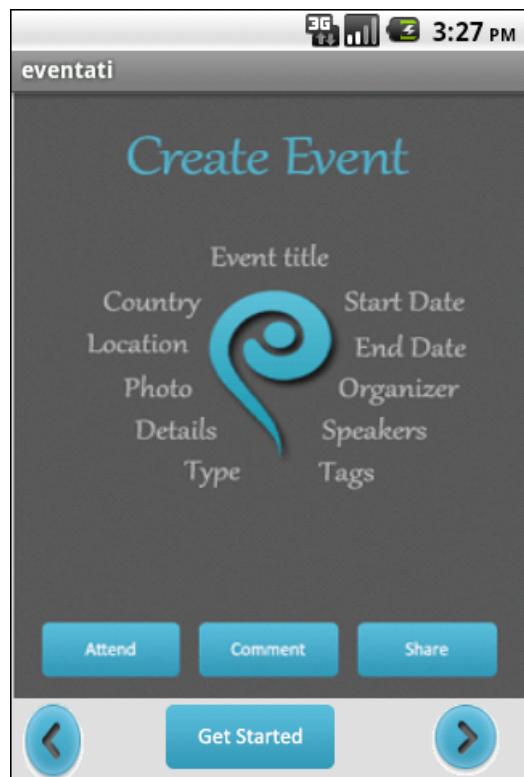


Figure 8.16: Slide 2 Of Mobile

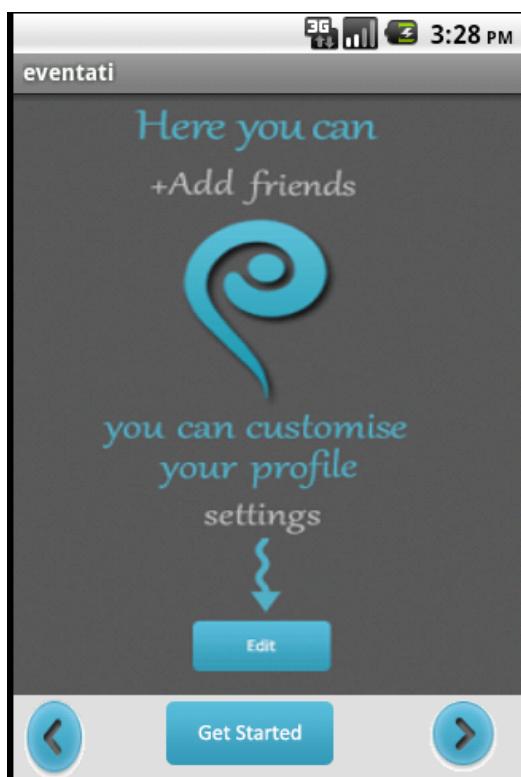


Figure 8.17: Slide 3 Of Mobile



Figure 8.18: Slide 4 Of Mobile

Chapter 9

Conclusion and future Work

In this project we have presented "Eventati" project components and the role of each component. "Eventati" components are: Web Component, Mobile Component, and Design Component. Briefly, each component's functionalities and specifications are as follows:

- **Web component:** an essential part in this project. It enables users to utilize all the functionalities provided by the project, such as create and share event, also allowed users to follow friends. "Eventati" can suggest friends for the users. Briefly, this component is the back bone of the project . This component have more than step and part to reach to the final view. We specified our aim (make social event directory) and work to activate it to achieve what we need (don't miss event). We specified our requirements and determined functional and Non-functional parts of them, started to make our project analysis and after that build our back end design at the time we have general view of our project shape after developing after that we make our front-end design and connect the back end design with front end design.
- **Mobile Component:** Second essential component of our project especially after the revolution of mobile world and mobile technologies, mobile became an important device to everyone, from here our idea of mobile app are stemmed. Mobile component can make approximately all functionalities of "Eventati" project. User can make, and share events, follow friends, suggest friends, and follow other events. Difference between mobile and Web applications is the limitation on mobile applications where users don't have the ability to create event only create it on web application.

Introduction acts as a bridge that transports readers from their own lives into the “place” of our analysis, but **conclusion** can provide a bridge to help readers make the transition back to their daily lives.

- **Design Component:** “Eventati” project designer use special technologies to reach to the final pretty design of both the Web and Mobile applications. Facing challenges of limited mobile resources, screen resolution, and the many features of the web site to present the user with interesting and amusing interface is the ultimate goal we have proudly achieved. Unifying Web site, Mobile Application, documentation,presentation, and all graphics related work within “Eventati” template was an interesting challenge.

Future Works include presenting a Firefox add-on that integrates with “Eventati”. Firefox add-ons now most of big social network have Fire fox add-ons to increase the availability of the networks. It helps users to be in connect to social networks during another work in Firefox browser Also, the increasing of tablets between users guides us to present a tablet version of “Eventati” so users can be connected to their events. We need to integrate maps deeply into our project. Transportation and on-line payment for events to allow users to pay to get event’s tickets and Book tickets are important values to be added to the project.

References

- [Arab Advisor Groups On Line Survey],22Jun2012 [URL] Available at[<http://d1glibrary.d1g.com/main/show/4239130>] [Accessed 24 March 2011].
- [avtec] ,[URL] Available at: [<http://avtecmedia.com/credit-union-marketing/social-networking/>] [Accessed 1 July 2012]
- [Flicker],March 2011 [URL] Available at: [<http://www.flickr.com/photos/99447525@N00/>] [Accessed 22 June 2012].
- [Hamid Shojaee],17 September 2008, [URL] Available at [<http://www.youtube.com/user/axosoft>] [Accessed October 2011] .
- [pyhton], [URL] Available at [<http://www.python.org/doc/essays/blurb.html>] [Accessed 27 June 2012].
- [Adrian Holovaty and Jacob Kaplan-Moss] ,26 June 2008, [Book] Available at [The Definitive Guide to Django Web Development Done Right, Second Edition] [Accessed 27 June 2012].
- [MVC], [URL] Available at [<http://media2mult.uos.de/pmwiki/fields/ame10/index.php?n=PHPTrax.MVCPattern>] [Accessed 27 June 2012].
- [James Tauber],10 October 2011, [URL] Available at [<http://pinaxproject.com/>] [Accessed 27 June 2012].
- [El-wasat], 25 May 2011 [URL] Available at:[<http://www.el-wasat.com/portal/News-55620338.html>] [Accessed 3 August 2011]
- [Django Application], 23 May 2011 [URL] Available at:[<http://www.programmersbook.com/page/39/Activate-the-Admin-Full-Web-Framework-Python-Django-Tutorial-VIII>][Accessed 30 June 2012].
- [Viktor, ShoutEm CEO], 7 February 2011 [URL] Available at:[<http://blog.shoutem.com/author/viktor/>] [Accessed 30 June 2012].
- [book keeper], 23 May 2011, [URL] Available at:[<http://www.programmersbook.com/>][Accessed 30 June 2012].
- [Richard Bateman], 21 August 2011 [URL] Available at:[<http://developer.appcelerator.com/devlink/profile/>][Accessed 30 June 2012].
- [Appcelerator,Inc.], 17 April 2012 ,[URL] Available at:[<http://en.wikipedia.org/wiki/Appcelerator-Titanium>] [Accessed 30 June 2012].