

2015

Eventure

TECHNICAL DOCUMENTATION

MATTHEW BOROCZKY
MICHAEL KONG
LOO YEE LIEW
RUIXI HE
CHEE YEONG LIM
SANDON JOUBERT
JUNYAN FAN

Table of Contents

Preamble	4
System Overview	5
Vision	6
Glossary	6
Problem statement	7
The problem:	7
The affected:	7
The impact:	7
The solution:	7
Position statement.....	9
Target Market:	9
Unlike competitors:	9
Our Product:	9
Marketing and advertising	10
Advertising and Marketing Strategy	10
Competitors and alternatives	12
System Structure.....	13
User environment	14
Scope and limitations	15
Constraints	15
Cost and Pricing	15
Quality Ranges	17
Procedures and Priorities.....	17
Documentation Requirements	18
Operational requirements	18
Technologies.....	19
Software	19
Development Tools	20
Hardware Specifications	21
Front End Specifications	21
Back End Specifications	21
System Architecture	22
Sub-system Relationship and Interaction Diagram.....	24
System Flows	25
System User Interface Design	28
Website Design	28
Mobile Design	32
Database Design.....	38
Database Schema	38

Data Dictionary.....	38
Third-party Libraries	47
Testing Plan.....	49
Testing inaccuracy	49
Techniques.....	49
Future development.....	50
Sensitivity	50
Assertion	50
Test cases	51
Black Box and White Box Testing.....	51
Testing tools	52

Preamble

This document records the technical information that pertains to the Eventure Conference Management System. All information within this document is primarily written for future iterations and developments, so that other developers and interested parties may be able to understand the structure and technical intentions of this system. Information within this document is preliminary and subject to change.

System Overview

Eventure is a conference and event management system that is seeking to integrate event organization capabilities, conference paper reviewing and in-event support. Eventure delivers an online web application and iOS mobile application that encompasses these capabilities to provide a guided process from event creation to in-house support.

The system includes features such as booking ticket, seat selection tools, seat release criteria, assigning speakers to rooms in a venue as well as an entire venue builder tool allowing the drawing of the venue without the use of third party applications.

Eventure allows users to create events and conferences. The difference between events and conferences is that conferences open paper reviewing features to the event organizer. In the paper reviewing features, users in the system who are skilled in particular areas can be invited to review relevant papers ensuring the integrity of the reviewing process. The paper management features will include reviewing features on papers, manual and automatic checking of conflict of interest, customer setting options such as determining minimum and maximum difference in papers reviews and much more.

The system also features a venue builder that allows users to build and design venues. Once the venue is created, it can be reused by other future events and conferences. Pictures, maps and available resources can be added to each created venue in the system to assist event organizers in choosing the right venue for their events.

Eventure will also be providing functionality to be used within the event itself such as: a mobile app that contains a map, has the ability to locate others and add them on LinkedIn and Facebook, has the ability to alert security if attendees are in need, has the ability to use phone ticketing system (iOS Passport) to hold ticketing information on phone and allow creation of a personalised timetable to sort out different performances/presentations.

These are just are few of the features we will provide that will allow us to cover a broad target market by covering numerous functionality and eliminating the need to use multiple application for the smooth running of the event. "We are with you step by step from the creation of your event to intra-event maps and timetable through to the post-event financial analysis and statistical accumulation."

Vision

Eventure is a tool we envisage as being a complete tool for both conference organising and events. Other tools are proficient at either conference management in terms of papers submission and review, while others are proficient at the managing and organising of upcoming events. Eventure combines these two worlds into one unique application. Not only are we concerned with the planning of conferences and general events but we are also concerned with ways we can improve the experience of event and conference attendees.

Glossary

Event Organiser - Individuals at the top of the hierarchy within the event. They have the ability to create and change all details about the event.

Conference - a formal meeting for discussion. When considering conferences, the system is not limited to small academic conferences. It should support the management of large conferences such as Apple's WWDC conference in which thousands attend (~ 5000)

Event – An event is similar to a conference but has functionality for paper reviewing is disabled.

Session - Sessions are held within conferences or events. One event/conference must have one or more sessions.

Author – Refers to an author of a paper. Also refers to event speakers since authors are potential speakers.

System – Refers to the Event Organiser and its subsystems.

User – Refers to any user of the system.

COI – Conflict of Interest

Problem statement

The problem:

It is difficult to find a tool that covers a large basis of planning, management, organisation and review of Conferences and general events, and covers it well.

Each application only covers at most one part of conference/event management, organisation and review (MOR) well, or covers and even spread across majority (not all) of areas involved in conference event MOR but does not cover it well enough/in enough depth/have a low level implementation. An example of this would be a software providing the ability to purchase tickets and ticket type but not having the ability to select seats using a GUI or having seat release conditions to ensure the company doesn't lose seats and profits. "Your profit is our profit, we both desire to optimise all aspects."

The affected:

Event and conference organisers, individuals wanting to create an event, event/conference attendees, event/conference presenters/performers, the public looking at attending an event.

The impact:

Companies would hire developers to create a venue on their website with seats and seat types to enable customers to be able to select seats for their venue. They would then have to use another third party software to manage the event and ticketing; an example of this would be EventBrite. Another third part software must then be used (such as EasyChair or ConfTool) to allow the conference proceedings such as paper submission and review to be allowed. Additionally no management tools, (such as EventBrite),(from our understanding), have the ability to purchase tickets for events and select the attending performances as there may be concurrent performances/presentations at one. In addition to this timetabling of the event is being provided. These tools are not provided by software we have found and therefore we can conclude developers would need to be hired to implement this functionality. Finally we must consider that developers may possible need to integrate these system, although some may be able to be used separately, integration is desirable to increase usability and efficiency, however this will incur greater costs from the company.

The solution:

1. To hire developers to Develop and integrated software solution and host it on the company's server.
2. Another solution would be to pay for all third party software and use it stand alone (note: some functionality will not be provided)

3. Another solution would be to pay for all third party software and pay developers to integrate the third party applications to achieve an integrated system (note: some functionality will not be provided)

4. The final solution . would be to pay for all third party software and pay developers to integrate the third party applications to achieve an integrated system and to get developers to build the functionality not included in the third party applications.

Position statement

Target Market:

Event and conference organisers, individuals wanting to create an event, event/conference attendees, event/conference presenters/performers, the public looking at attending an event.

Unlike competitors:

Unlike ConfTool and EasyChair we plan to have our services available online and we will not limit ourselves to just paper reviewing and general conference management.

Unlike EventBrite we will provide advanced features such as seat selection, seating release conditions and a venue building tool. We will also provide a in-house system which will allow timetabling, networking and security measures to be in place.

Our Product:

Eventure will be using the Software as a Service (SaaS) model which will reduce the need for the software to be installed locally on their servers and eliminate them to pay for the entire software outright. Using SaaS will reduce their upfront payment (to nothing) and increase our profitability (the ones hosting the software) as we will require a dividend on the purchased ticket prices. This will also allow our software with limited functionality to be used for free with events that have free tickets and less than 200 capacity for the event (for events not conferences). In return we will offer our Eventure which integrates functionality that would usually require numerous third party software to be used. Not only that but we do not only strive to cover a spread of functionality but to cover each functionality well to ensure the highest degree of practicality and usability.

Marketing and advertising

Marketing and advertising is a very important aspect of any product. A product may be really good but if it does not have good advertising, the product may fail.

Conversely, a product which may not have many significant advantages over other similar products, but has a good marketing campaign, can lead the product to success.

Hence, we believe marketing and advertisement is important. Below is a list of our marketing and advertising strategy.

Advertising and Marketing Strategy

1. Target market

This is the important part in Marketing Strategy, because every strategy needs to understand the target market. We are targeting event organizers, academic and industry professionals, and the general public. Event organizers will be one of the main targets because they will be creating the events for participants to join. Academic and industry professionals will be the leading group in the conferences because they would include people at the forefront of major technological developments and research. The general public is another target group because this group will be the core attendees of any event.

2. Marketing methods

The number of users who will use our system is a good indicator to evaluate whether our system is successful or not. To attract users, we have a list 3 methods:

- a. Limited free trial. New users can experience 3 months of free trial on our system. This will give them time to explore, learn and get familiar with our system.
- b. Hunger marketing. We create an advertisement that states if the user upgrades their membership within the 3 months of limited free trial, you can they will get an extra 1 year membership for free. This method attracts users to invest in our system for a longer period of time.
- c. Building a good brand image. Brand image is important for a new product because a good brand image promotes the integrity of the product. Therefore, people are more willing to trust products with a good brand image. To build a good brand image, we can cooperate with established enterprises because such enterprises have good brand image. Therefore, if established enterprises use our product, people will view our product as trustworthy.

3. User retention

Products are made to serve people. If the service that the product provides is lackluster, users will not use our product. We list two strategies to retain users:

- a. Product improvement. Since Eventure is a software, existing features will continually improve and new features will be developed. These will assist in user retention to meet the current needs of users.
- b. Loyalty programs. If a paying user has used our product for 2 years and has created 10 events within that time span, he can get discount in for the next year's membership or he will get point which instead of some money to attend some events.

4. Attracting users to upgrade to a paid account

A balance the free functionality and paid functionality is key. If the free functionality provided allows the user to perform all tasks within the system, they will not find the need to upgrade. However if the free functionality stifles the user's ability to experiment with our system's basic functionalities, they will not use our product either.

Competitors and alternatives

As with any product, we have competitors and alternatives. If we want our product to stand out, we need have an edge over the other competitors. The following list contains the three most popular competitors and an analysis on their advantages and disadvantages.

1. Eventbrite

Eventbrite is a website that allows people to find and create events in different countries. Being a popular event management system, it has many comprehensive event features. However it does not have strong support for conferences which require paper reviewing and validation. Eventure has an advantage over Eventbrite here because we have features to support both events and conferences.

2. Easy Chair

This is a conference manager system that has support for paper reviews and validation. While It is a good conference manager, it has some disadvantages:

- a. Lackluster user interface. When users first enter the Easy Chair website, there is nothing attractive about the site due to the plain pastel colour scheme.
- b. Requires download and installation. Since the emergence of software as a service (SaaS) delivery models, many users prefer systems which require no installation and convenient access. While Easy Chair does allow users to host the Easy Chair software on their own servers, it does not provide the convenience of SaaS delivery models which Eventure will be following.

3. Conference Alerts

This website is a conference aggregator and web portal. Users can find many different conferences on this website, however when users select a conference they will be redirected to a webpage that list basic info about the conference and a web link to an external web page for that particular conference. So conference alerts is simply a web portal for conferences so there is no real conference creation capabilities on this system.

In conclusion, we have the following advantages over the aforementioned systems:

1. Eventure has features to create both events and conferences, giving users the flexibility to choose a basic event or a more sophisticated conference.
2. By focusing on UI and UX design, we aim for an aesthetic and usable system and web application.
3. Deploying Eventure as a SaaS, users have the capability to access our system from desktop or through the mobile application.

System Structure

The structure of the system can be broken down into 4 distinct components:

- **Online website**
The main interface of the system. It consists of a series of webpages that allows the users to access the features of the system. All features are available on this component except the in-house subsystem which will be handled in the iOS application.
- **iOS Application**
The secondary interface of the system. It is a mobile application that provides a subset of the features available on the webpage such as purchasing a ticket for an event and managing user profile. The iOS application will also have additional functionality by providing the user with an in-house subsystem which provides features such as providing venue maps and check-in services.
- **Application server**
The application server is the intermediary between the database server, the iOS application and the online website. It performs inserts and updates on the database when users add or change items on the website or iOS application. It also performs queries on the database, processes the data and returns the result to the website or the iOS application.
- **Database server**
This component stores all the data that is considered persistent in the system, such as user emails, submitted papers and event details.

User environment

The user environment for the system is the standard ADSL internet connection, web browsers such as Google Chrome, Safari, Firefox, Internet Explorer etc. To support web browsing on personal computers and laptops, operating systems namely Unix, Microsoft Windows, OS X and Linux will act as the fundamental operating environment.

In this project we will be developing a mobile application complimentary to the system, the mobile platform chosen is iOS. Network connection for the mobile device is also essential for downloading the application and using it for real-time updates.

Lastly, e-mail access is also required in order to fully utilize the notification functionality of the system. Users may need to have an active existing e-mail account or will be asked to create one.

Scope and limitations

The team will be using Amazon Web Services (AWS) for server side services. AWS offers a broad set of global computation solutions, storage solutions, database solutions, analytics solutions, application solutions, and deployment services that help organizations move faster, lower IT costs, and scale applications.

As the AWS calculates cost and pricing by every input and output, the development team has to minimize interaction with the AWS during development. Developers have to run the test servers on their local machines for testing and fixing bugs. When features are tested and debugged, the team will push those features to the server.

Although there are 7 members in the team, the team lacks experience in back-end development. Therefore, it is a challenge for the team to develop a scalable, maintainable and efficient system.

Constraints

The Event Management system and its requirements are limited by the following constraints:

- Network - The performance of the system is dependent on the performance of the network connection it is on. The requirements defined in this document assume a standard ADSL network connection.

Cost and Pricing

- Cost for user

The Event Management system will be available for free to the general public. However, the system will also provide a paid subscription service, where a user will pay a small amount of monthly fee to access additional features of the system, such as:

1. Creation of new events
2. Making use of the Conference Management features (e.g.: paper submission and paper reviewing)

As the system is not being marketed as a product, there will be no additional charges (such as licensing) to potential vendors.

- Cost for owner

AWS offers a simple, pay-as-you-go pricing approach for over 40 cloud services. With AWS the team will only pay for the services that the team needs. Since there are no long-term contracts, the team does not have to worry about dependencies, complex licensing or termination fees.

AWS offers cloud services with a free tier available for 12 months. On a monthly basis on the free tier, the services that the team will be using are,

1. 750 hours of EC2 instances with 1 vCPU and 1 Gigabytes of RAM;
2. 5 Gigabytes S3 standard storage with 20,000 get and put requests;
3. 750 hours of RDS instances with 20 Gigabytes storage and 10 million Inputs/Outputs;
4. 1 million API calls;
5. 750 hours of Elasticsearch instances.

The aforementioned specifications are sufficient for the team to run the project for one year unless there is an exponential growth of users for the project. Different services will have different pricing schemata. For example, if the project is too large for the free tier EC2 compute engine, the team will have to pay USD 0.02 per hour* for better server instances, or if the 5 gigabytes of storage is not enough to store all the papers and other files, the team will need to pay USD 0.03 per gigabyte for extra storage for the first terabyte*.

The cost and pricing of AWS above is correct as of 30 September 2015. The price might change as the follow the Moore Law, the price for cloud services will decrease dramatically over time. It will be more suitable to refer to the pricing sites in AWS sites all the time.

*More information about pricing for AWS can be found at:
<https://aws.amazon.com/pricing/services/>

Quality Ranges

The following points describe the quality ranges and expected performance benchmarks for the Eventure system:

- Availability – The service will be available 24 hours a day, 7 days a week, with 2 hours downtime a week during non-peak hours for scheduled maintenance.
- Usability – The service will be easy to use for both novice users, who are expected to have only small amounts of technical knowledge, and for the expert users, who are expected to have good understanding of the system and its components.
- Maintainability – The system is easily changed and configured without the need to reconstruct or redeploy the system.
- Accessibility – The service will provide help for the users, in forms of online help and user and administrator manuals.

For additional quality ranges can be found in the Non-Functional requirements section of the Requirements document.

Procedures and Priorities

First of all, the team will design the application web interface. Using the created interface, the team will determine the data required to be stored in the database. After most of the requirements for the web is complete, the team will start development on the iOS application because it is an extended feature of the whole system and not a core component of the system.

The team will first focus on core requirements, and then move on to main requirements. The team will also try their best to implement requirements in stretch goals and dream goals. Details on the priority of the requirements and the order they are to be implemented can be found in the Software Requirements Specification document.

Documentation Requirements

The following documents are required for the full system release:

- Technical Documentation – This document describes the development process and the structure of the system. With this document, future developers are able to understand the system's technicalities and make changes to the system.
- User Manual – This document serves to help users understand certain parts of the system that may have a complicated process and describes features that not all users might not use on a daily basis. A downloadable copy will be available from the Eventure website.
- Online Help – This help is provided in two ways. Firstly, it is a set of FAQ's on the Eventure website or the Eventure mobile application. Secondly, help is also provided via tool-tips, icons and a standard interface on the Eventure website and the Eventure mobile application.

Operational requirements

To access the Eventure website, users must meet the following requirements:

- Web browser
- Network connection

To access the mobile application, users must meet the following requirements:

- iPhone or iPad running on iOS 8 and above
- Network connection

Optional/extended operational requirements:

- Printer to print self-print tickets or details on events for reference
- Label printer for organizers to print event labels
- QR scanner to scan user's QR code tickets

Technologies

Software

- PHP – PHP is an open-source, server-side scripting language. This language is used in Eventure because it is one of the most used server-side scripting languages. Therefore, help resources online are widely available. In Eventure, PHP will be used for three main purposes: processing data that was queried from the database; storing given data into the database; and tracking user sessions.

Reference: <http://php.net/>

- SQL – Stands for Structured Query Language. This is a default query language for many databases available. Eventure will use this query language to pass data between the PHP scripts and MySQL database.
- MySQL – An open-source relational database developed by Oracle Corporation. This software provides powerful database capabilities that enable it to rival systems such as Oracle and DB2 in terms of features. Although MySQL may not have the processing capabilities of more powerful systems such as Oracle and DB2, it is a sufficiently good, open-source system for use in the Event Organizer.

Reference: <https://www.mysql.com/>

- SQLite – This is a popular embedded database that will be used in the Eventure iOS application. The embedded SQLite database will be used to store data that is personal to each device such as user name and ticket details.

Reference: <https://www.sqlite.org/>

- HTML5/CSS3 – HTML5 and CSS3 are the new standards for web development since 2014. HTML is a markup language used by web browsers to render content and CSS is a style sheet language used to define how the HTML markup is presented.
- JavaScript – A commonly used programming language in website and web app development. It is supported in many web browsers natively and it is used to dynamically manipulate the Document Object Model (DOM) of HTML documents and the CSS styles of web pages.
- JSON – JSON is a data structure that is lightweight and supported by many programming languages. It is commonly used as a means of interchanging data between applications. In the Event Organizer, it will be used as a means of communicating between the web client/iOS application and the application server.

Reference: <http://www.json.org/>

- Objective-C – A programming language largely used in developing OS X and iOS applications. Currently there are many Objective-C libraries written specifically for iOS applications which reduces the amount of time in writing and testing code.
- Swift – A programming language developed by Apple Inc. for developing applications for iOS, OSX and watchOS. Since Swift's language syntax is simpler than Objective-C's, Swift speeds up application development time. Furthermore, it is cross compatible with Objective-C, which gives Swift applications access to existing Objective-C libraries.

Reference: <https://developer.apple.com/swift/>

Development Tools

The following development tools will be used in the development of Eventure:

- Google Chrome for web
- Mozilla Firefox for web
- Sublime Text
- Visual Studio Code
- Atom
- Xcode IDE for iOS
- GitHub versioning system

Hardware Specifications

Front End Specifications

Organizers

- QR code scanner
- Label printer

Users

- Personal computer
- Laptop
- iPhone
- iPad
- Printer

Back End Specifications

Host: Amazon RDS (database services)

Usage: Managed Relational Database Service for MySQL, Postgres, Oracle, and SQL Server

Support:

- 750 Hours of Amazon RDS Single-AZ db.t2.micro Instance usage
- 20 GB of DB Storage: any combination of General Purpose (SSD) or Magnetic
- 20 GB for Backups (with RDS Magnetic storage; I/Os on General Purpose [SSD] are not separately billed)
- 10,000,000 I/Os

Host: Amazon EC2 (server processing services)

Advantage: Resizable compute capacity in the Cloud

Support:

- 750 hours per month of Linux, RHEL, or SLES t2.micro instance usage
- 750 hours per month of Windows t2.micro instance usage
- For examples, run 1 instance x 1 month or 2 instances x half a month with 750 hours

Host: Amazon S3 (storage services)

Advantage: Highly scalable, reliable, and low-latency data storage infrastructure

Support:

- 5 GB of Standard Storage
- 20,000 Get requests
- 2,000 Put Requests

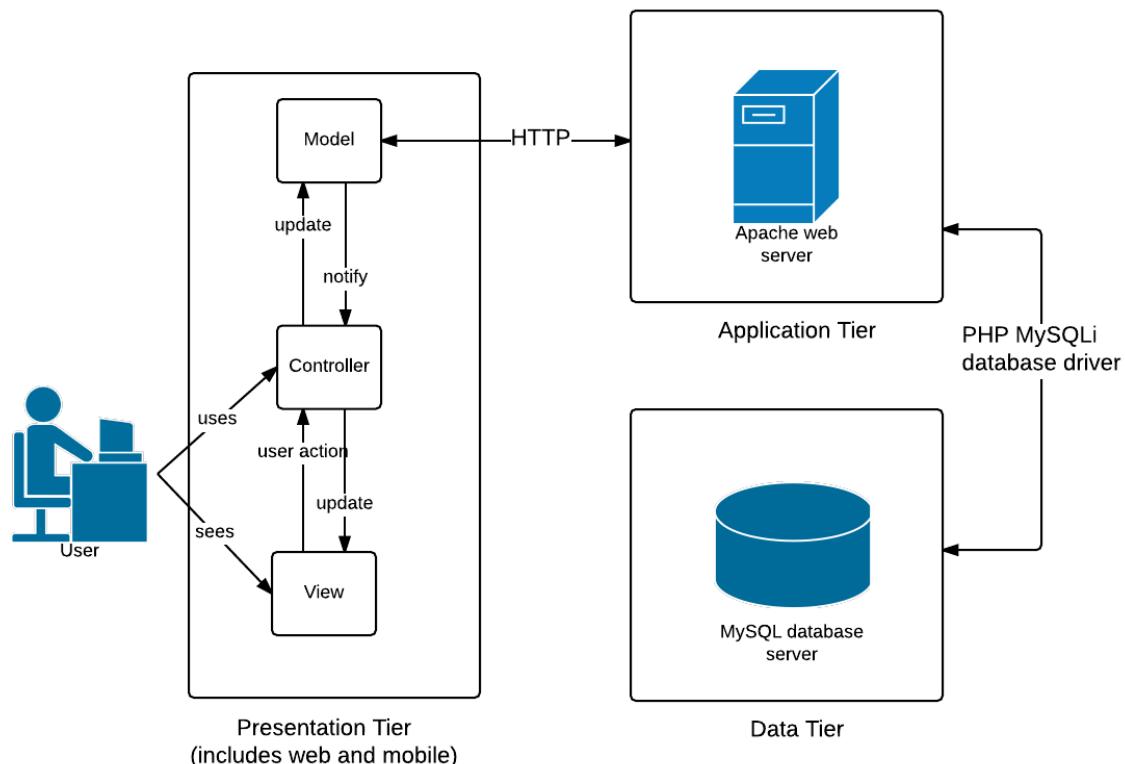
System Architecture

The entire physical system is based on a 3-tier client-server architecture, namely:

- Presentation tier, which presents the user interface and is the primary source of interaction between the users of the system and the system itself. This tier performs some data computation.
- Application tier, which is the coordinator of the application and performs most of the processing of the system.
- Data tier, which handles all data storage, management and retrieval.

The presentation tier can be further decomposed into a Model-View-Controller architectural pattern.

A graphical representation of the physical system architecture is shown below:

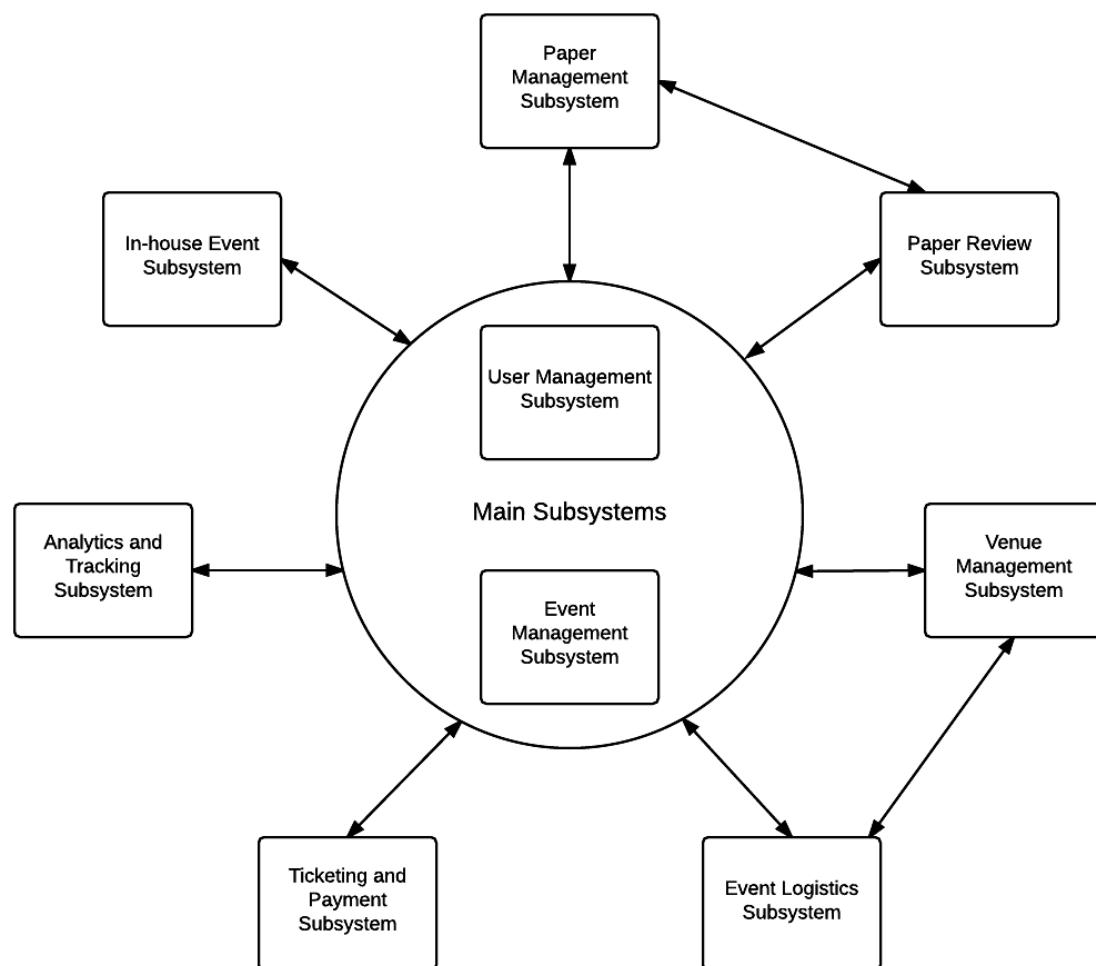


The logical aspect of the system is based on a repository architecture where the User Management Subsystem and the Event Creation Subsystem are the two subsystems on which the other subsystems are based on. With this architecture, the two main subsystems can be built first and the rest of the subsystems can be added in later prototypes of the system.

The nine subsystems of the Event Organizer are:

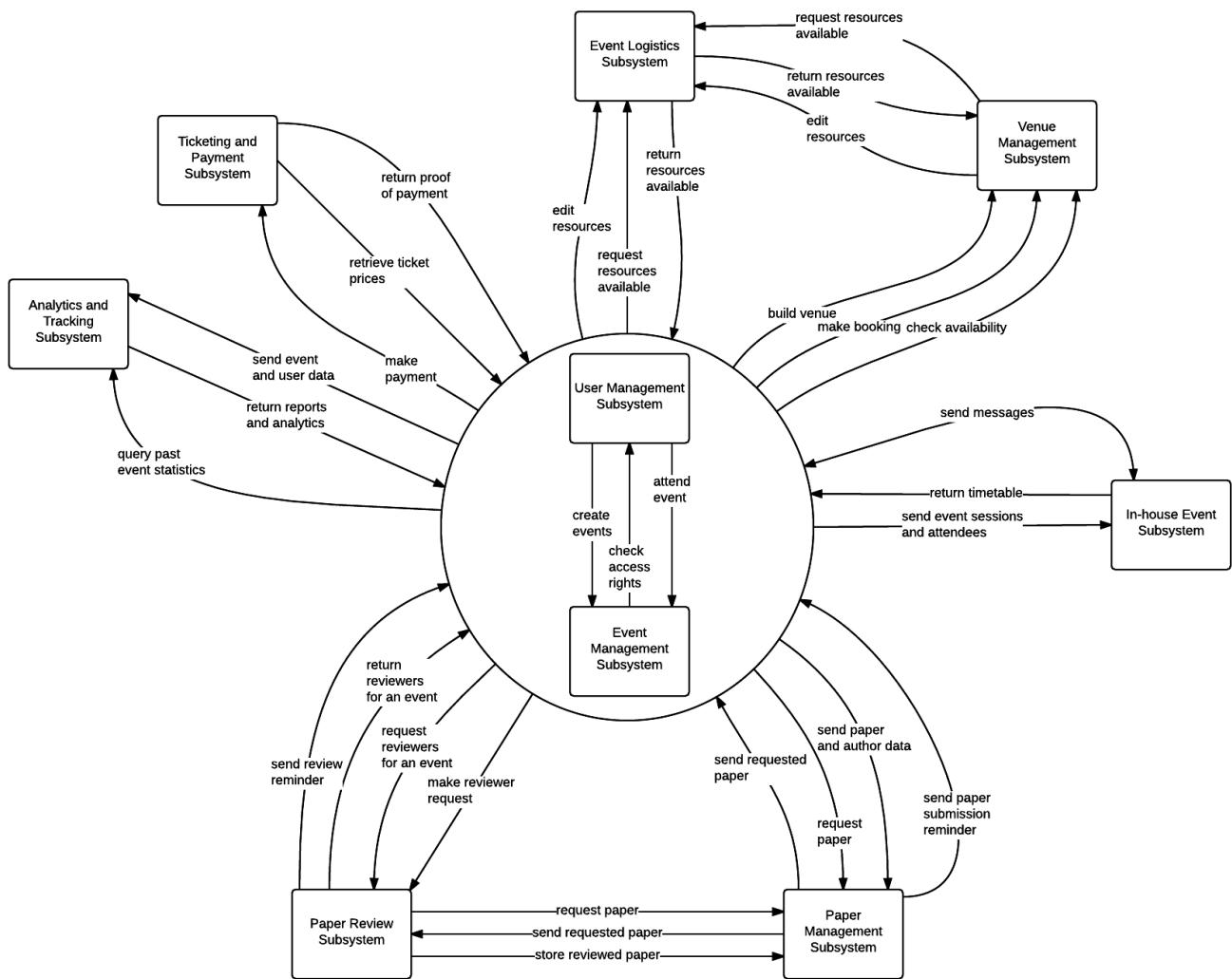
- User Management Subsystem that manages user profiles and access rights.
- Paper Management Subsystem that manages paper submissions, deadlines, sorting, filtering and storage.
- Paper Review Subsystem that manages reviewer requests, provides features for review procedures and protocols, and tracks reviews of papers.
- Event Management Subsystem that allows the creation of events, choosing of venues and setting of ticket prices.
- Venue Management Subsystem that allows venue building via a “room builder” feature.
- Event Logistics Subsystem that manages the various resources such as microphones and projectors during an event, and allow users to specify meal preferences.
- Ticketing and Payment Subsystem that allows users to purchase tickets.
- Analytics and Tracking Subsystem that tracks various aspects of the system such as the demographics of the event attendees and produces financial reports of events.
- In-house Event Subsystem that contains timetables of events, provides venue maps, facilitates check-in and sends post-event questionnaires to event attendees.

A graphical representation of the logical system architecture is shown below:



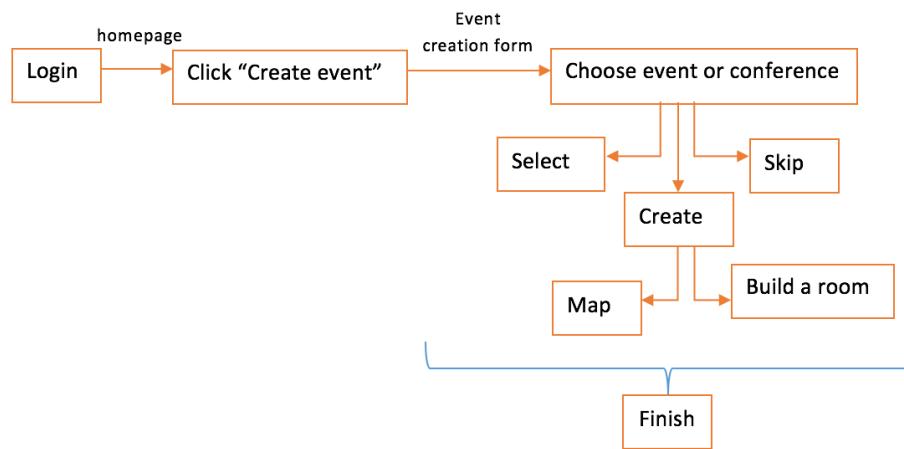
Sub-system Relationship and Interaction Diagram

The diagram below describes the interaction and relationships between the 9 subsystems of the Eventure Conference Management System.

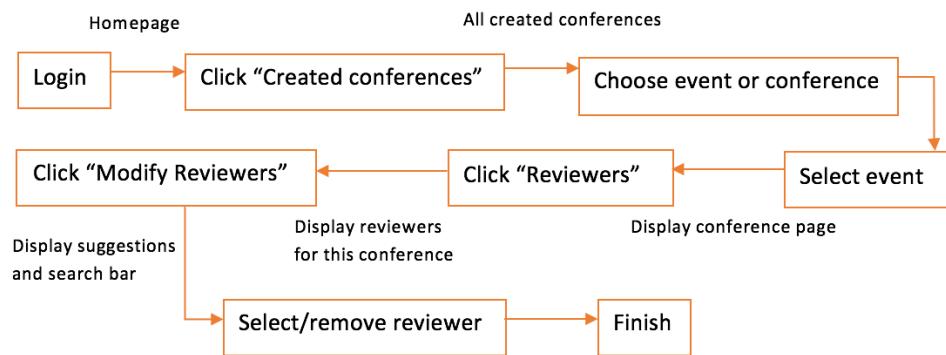


System Flows

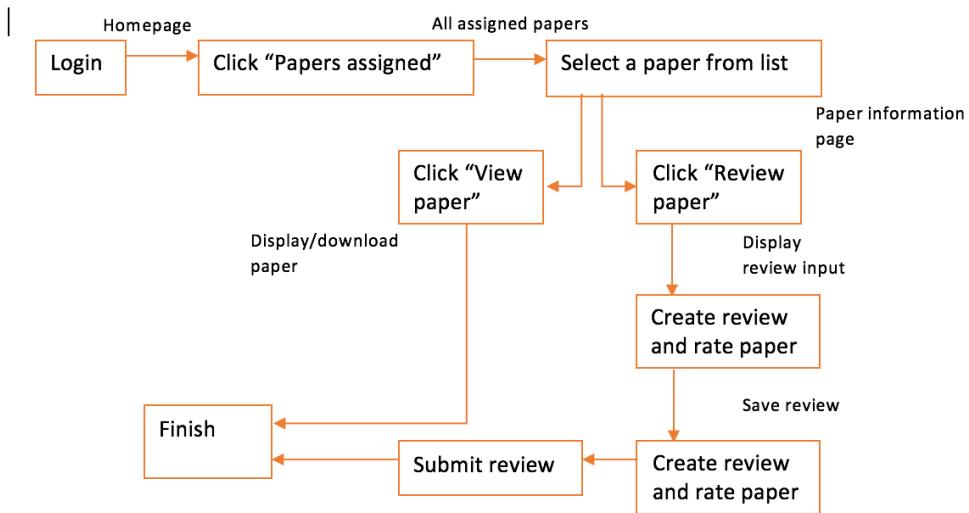
Conference creation flow



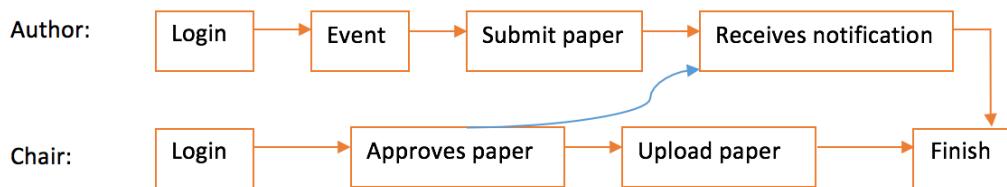
Select reviewers



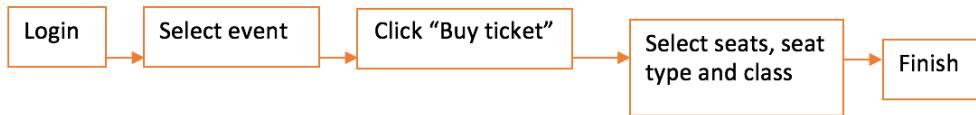
Review paper (Reviewer)



Submit paper



Buy ticket

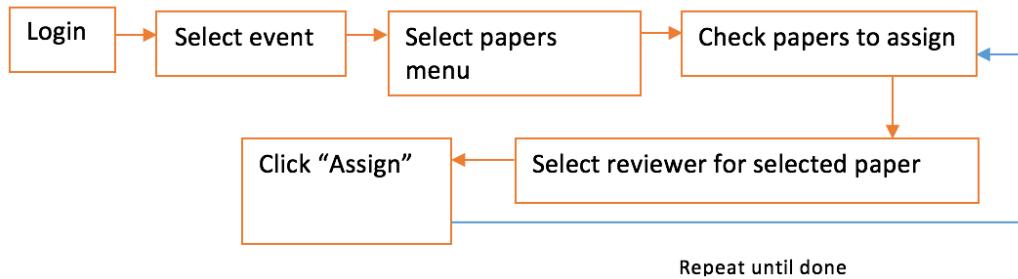


Upgrade account



Assign paper to reviewer

Event manager:



Check COI

Event manager:



System User Interface Design

Website Design

Header

The header displays the Eventure logo, a search bar to search for events or conferences and the user's account panel. The user's account panel will display the profile picture and "My account" button, that allows user to navigate to their profile to view personal details as well as make changes. There is also a log out button for logging out. The design of header bar uses a turquoise shade that will be widely used in this website to create an interface that is easy on the eyes.



Figure 1.1 Header bar for users who are logged in



Figure 1.2 Header bar for guests

Side menu

The side menu for guests and logged in users are distinctive from each other in terms of contents.

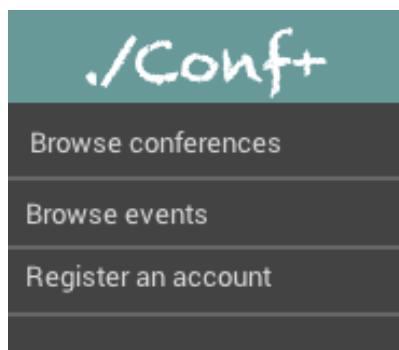


Figure 1.3 Side menu for guests

The side menu for users who are not logged in provides limited options in contrast to users who are logged in.

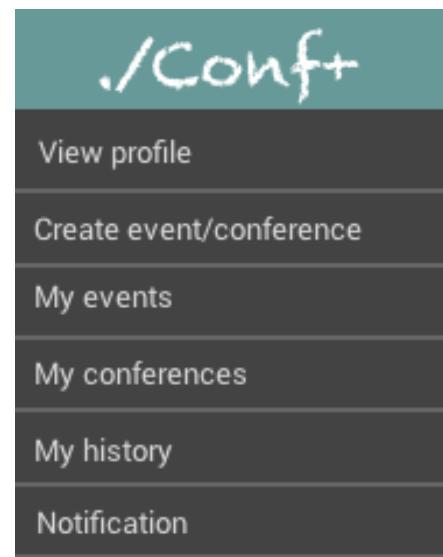


Figure 1.4 Side menu for logged in users

However, a logged in user sees a side menu with tailored options that allows them to view personal information and histories. For instance, "View profile" option navigates to a page containing user's personal details, "My events" or "My conferences" option navigates to a page where information of the user's attending events or conferences will be displayed, "My history" shows user's attended events or conferences, and "Notification" is where user

can be notified about event updates, event invitation, paper review requests etc. To “Create event/conference” the user will need to have an upgraded account, or the page will just simply provide an upgrade option.

Page content

Page content varies depending on the page and its elements, however the general page design consists of white backgrounds, simple dark grey text and hints of turquoise to create aesthetically pleasing interface. In order to adhere to design consistency, the color schemes and arrangements page elements are similar.

The guest main page is topped with a call to action (CTA) and a quick start button to guide users through the system and option sign up for an account. This page is filled with popular conferences and events, recently added conference and events list, conference and event tags to make browsing convenient and accurate.

The screenshot shows the homepage of a conference management application. At the top, there's a teal header bar with the text "Your very own conference manager" on the left, a search bar in the center, and a "Log in/sign up" button on the right. Below the header, a large banner features the text "Create your own conference or event today!" and a prominent green "Get started" button. The main content area is divided into several sections:

- Popular conferences:** A grid of four cards. From left to right: "Barcode of Life", "Discover, Build, Challenge.", "Student as producer", and "Design Indaba Conference". Each card has a "More.." link to its right.
- Popular events:** A grid of four cards. From left to right: "Laugh it out!", "12 Towers", "World Food Festival", and "TREAFEST". Each card has a "More.." link to its right.
- Recently added conferences:** A grid of four cards. From left to right: "Student as producer", "Mission Conference", "Media Histories: Epistemology", and "World Town Planning Day". Each card has a "More.." link to its right.
- Recently added events:** A grid of four cards. From left to right: "Gainesville Fashion Week", "TREAFEST", "Spingfest", and "Passatges". Each card has a "More.." link to its right.
- Browse conferences by tags:** A row of colored buttons for filtering conferences: Computing, Mass media, Designing, Science, Globalisation, Education, Business, Politics, Medical, and Environment.
- Browse events by tags:** A row of colored buttons for filtering events: Music, Fashion, Charity, Politics, Car show, Business, Art, Cultural, Sports, and Food.

Figure 1.5 Guest main page

When the user logs in, the main page is built with elements that is tailored for the user based on his/her history. For example, the user receives notification from events that they are going to attend, reminders on upcoming events or deadlines (paper submission and review), manage the user's own events or conferences, and also recommended events and conferences that may match the user's interest analyzed from user's history. Browsing by tags are also displayed here for convenient event/conference searching.

Your very own conference manager

Welcome back, Obama

Search Mv Account Logout

You have 2 new notifications

- Bell icon The event details of [Media Histories: Epistemology, Materiality, Temporality](#) has been updated. 15 minutes ago
- Bell icon [DDR 2013](#) requests your feedback on the event. 2 hours ago

Event reminders

- Clock icon In 3 days : Paper submission deadline for [Student as Producer](#) (13/1/2015)

Your conferences

		
Design Unite [ATTENDED]	DDR 2013 [ATTENDED]	Media Histories: Epistemo... [UPCOMING]

Your events

You don't have any upcoming or history events yet

Recommended conferences

			
Barcode of Life	Discover. Build. Challenge.	Student as producer	Design Indaba Conference

More..

Recommended events

			
Laugh it out!	12 Towers	World Food Festival	TREAFEST

More..

Browse conferences by tags

Computing	Mass media	Designing	Science
Globalisation	Education	Business	
Politics	Medical	Environment	

Browse events by tags

Music	Fashion	Charity	Politics
Car show	Business	Art	Cultural
Sports	Food		

Figure 1.6 User main page

Lastly, the event/conference page is made up of event details, event poster, get tickets or go to event/conference official website buttons, related tags, event/conference sharing methods (e.g. Facebook, LinkedIn, Twitter and so on) and also similar events/conferences recommendations.

Design Indaba Conference

Venue : Auditorium 1, Cape Town International Convention Center, South Africa

Time & Date : Wednesday, 25 February, 2015 - 08:45 to 17:00
Thursday, 26 February, 2015 - 08:45 to 17:00
Friday, 27 February, 2015 - 08:45 to 17:00

Description : The annual Design Indaba Conference in Cape Town features the best of global creativity on one stage. It is also simulcast live to cities around South Africa.

Every year the programme features a sterling lineup of international speakers from all corners of the world, coming together on the Design Indaba stage to present an up-to-the-minute survey on design and innovation from across the creative sectors. Surprise elements of dialogue, performance and music intersperse the speaker programme for an inimitable experience that makes Design Indaba what has been said by many to be the best creative conference in the world.

Tags : [design](#), [creative arts](#), [music](#), [performance](#)



[Get tickets](#)

[Official website](#)

Share via :

You might also like :



Barcode of Life



Discover. Build.
Challenge.



Student as
producer



Design Indaba
Conference
2013

[More suggestions](#)



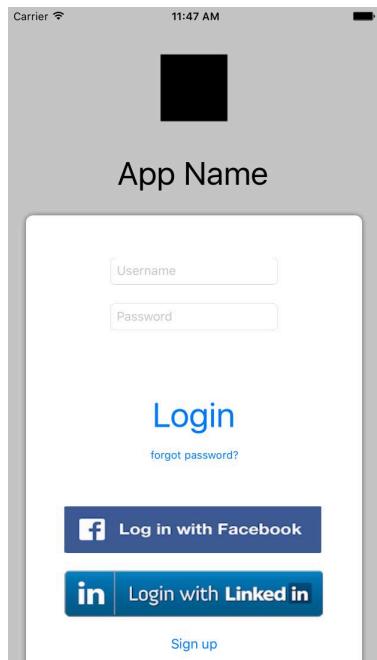
DDR 2013
Conference

Mobile Design

The mobile version of this system mainly focuses on event attendees, and their interaction between each other, or interaction with organizers. The designs displayed here are merely prototypes, and will be further developed.

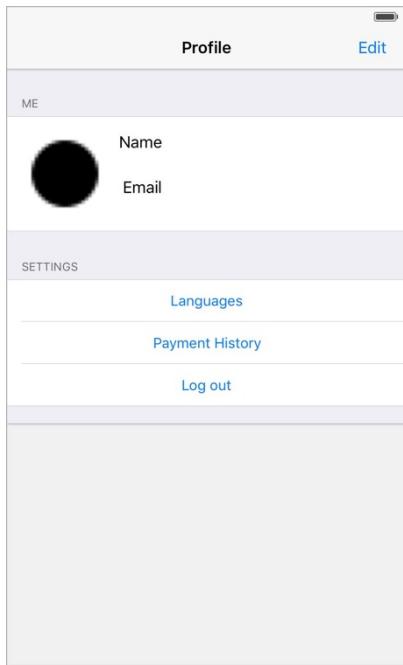
Login page

The login page is the pop up page when an application is launched, for users who are using the app for the first time, or have not logged in. They can either log in with their own created account, or choose to log in via Facebook or LinkedIn. Users who have forgotten their passwords can get help from the “forgot password” link, or sign up for an account if they have not done so. User who failed to log in will not be able to utilize the application.



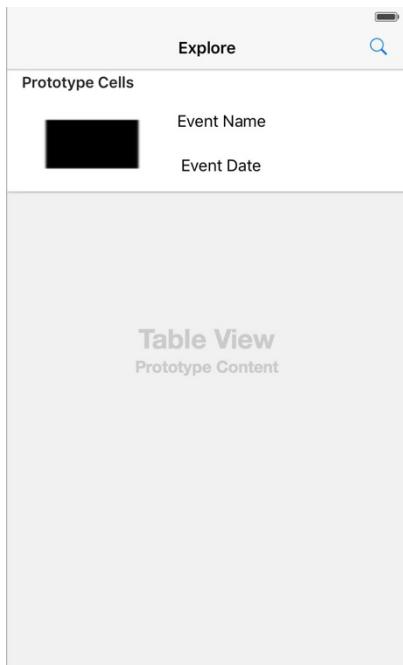
User profile

The user profile shows the user's name and email, allowing them to edit details with the Edit button on the right top corner of the screen. Apart from that, there are settings for users to customize the application by changing language settings, to view their own payment history, or log out from their profile.



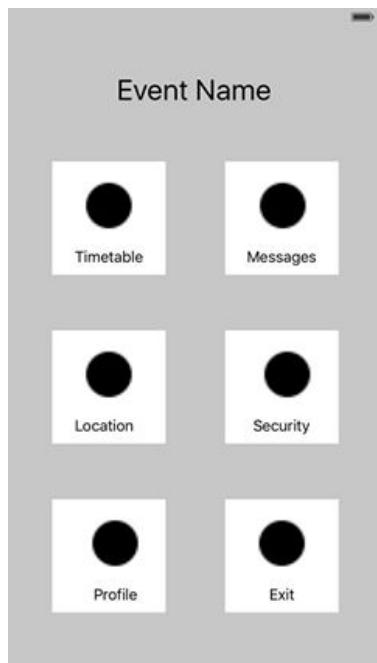
Explore events

Displays a list of events containing event name and event dates. By clicking on any of this cells will navigate to the event main page.



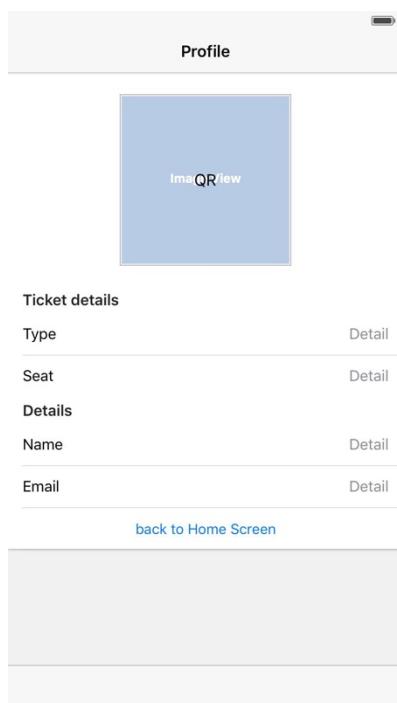
Event homepage

The event homepage contains icons that navigates to pages containing event information such as timetable for event programs, messages, location of the event, event security, event profile, or exit the event page.



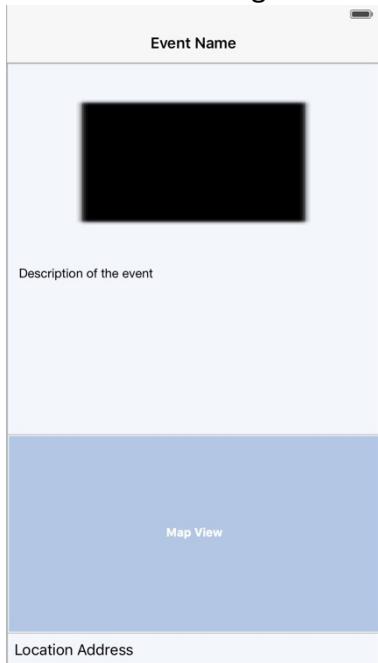
Event profile

This page displays the event profile that contains information that explains the event information including ticket details, and user details such as name and email, to allow organizer to check in attendees.



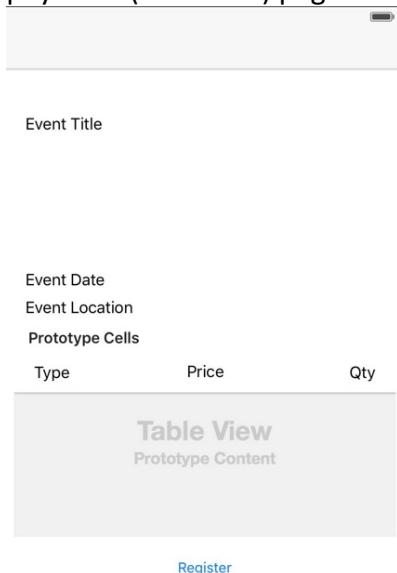
Event details

This is a page containing event details, containing event information such as event poster, event description (venue, time etc.) as well as the location aided with a map so that attendees can navigate themselves to the event location.



Event with tickets

This page is an extension from the event details page, navigated from tapping on buy ticket button. The page is made up of event details, and list of tickets available for the event. The user can select the quantity of tickets they would want to purchase here, and proceed to payment (if needed) page.



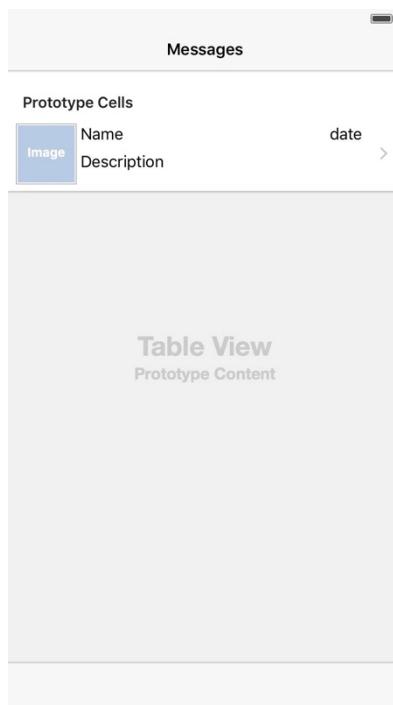
Timetable

The timetable is customized for user with their arrangement of talks or programs they want to attend to, making it easier for attendees to arrange their time while attending the event.



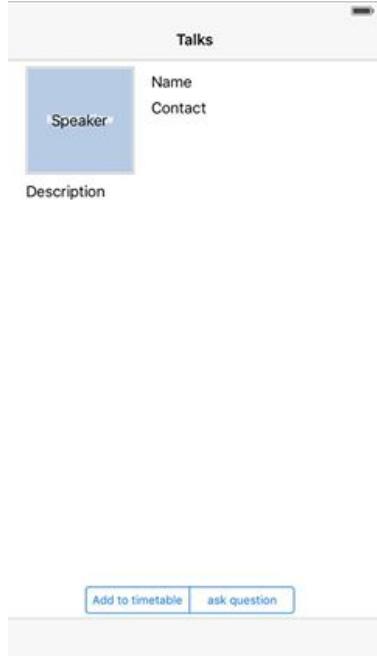
Messages

The messages page enables users to interact with one another, either communicate with other attendees of the event, or receive updates from organizer, or interact with the speaker when the talk is going on as a part of the event. The page displays the messages in a list containing the sender's image, name, the message details and the date the message is sent.



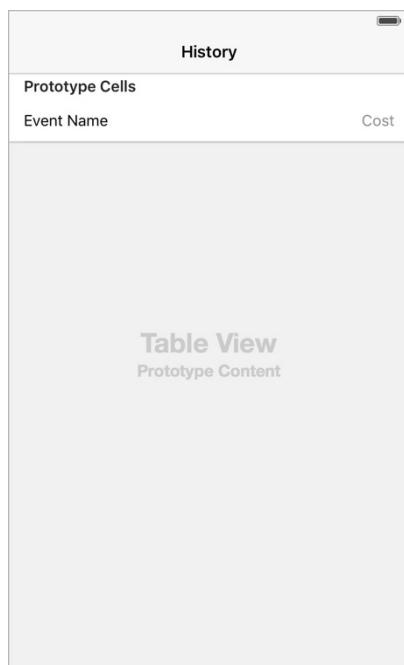
Talks or programs

A list of talks or programs offered by the event will be displayed in this page with the speaker's picture or the talk poster, name of the speaker as well as ways to contact the speaker. The description of the talk will be shown at the bottom of the list item.



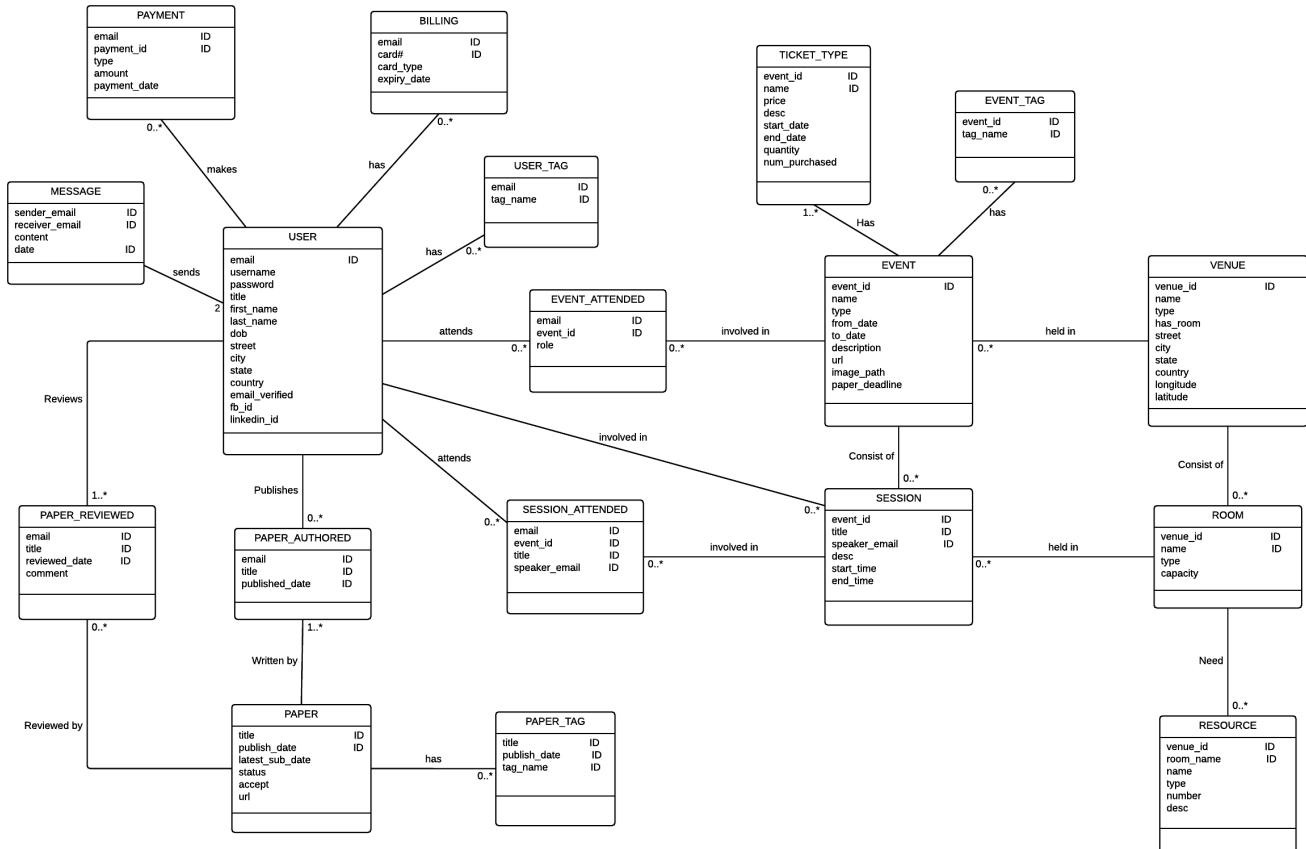
Event history

Tailored for users, shows their attended event history in a list. The cost label displays the amount the user spent on the ticket for this event.



Database Design

Database Schema



Data Dictionary

Table: USERS

Field Name	Data Type	Key	Null	Default	Description
email	TEXT	PK	NO	NONE	User's email address as the unique identifier
username	TEXT	CK	YES	NULL	User name of user which made by themselves
password	TEXT		NO	NONE	Password of user
title	TEXT		YES	NULL	The title of user (Mr, Miss, Mrs, Doctor)
first_name	TEXT		YES	NULL	First name of user
last_name	TEXT		YES	NULL	Last name of user
dob	DATE		YES	NULL	Date of birth of user
street	TEXT		YES	NULL	Street name of user's address
city	TEXT		YES	NULL	City that user stay
state	TEXT		YES	NULL	State that user stay

country	TEXT		YES	NULL	User's country
verified	BOOLEAN		YES	NULL	Whether the user account been verified by email
fb_id	NUMBER		YES	NULL	User's Facebook ID
linkedin_id	NUMBER		YES	NULL	User's LinkedIn ID
active	BOOLEAN		YES	NULL	Whether the user account been active or not
upgraded	BOOLEAN		YES	NULL	Whether the user account been upgraded or not
review	BOOLEAN		YES	NULL	Whether the user wants to be a reviewer or not

Table: USERS_TAG

Field Name	Data Type	Key	Null	Default	Description
email	TEXT	PK, FK(Users)	NO	NONE	User's email address as the unique identifier, references Users
tag_name	TEXT	PK	NO	NONE	Name of user tag

Table: COI

Field Name	Data Type	Key	Null	Default	Description
reviewer	TEXT	PK, FK(Users)	NO	NONE	Reviewer's email address as the unique identifier, references Users
author	TEXT	PK, FK(Users)	NO	NONE	Author's email address as the unique identifier, references Users

Table: CONVERSATIONS

Field Name	Data Type	Key	Null	Default	Description
conversation_id	INTEGER	PK	NO	NONE	Unique identifier of a conversation
name	TEXT		YES	NULL	Name of a conversation

Table: MESSAGES

Field Name	Data Type	Key	Null	Default	Description
message_id	INTEGER	PK	NO	NONE	Unique identifier of a message
sender_email	TEXT	FK(Users)	NO	NONE	Sender's email address as the unique identifier, references Users
conversation_id	INTEGER	FK(Conversations)	NO	NONE	Unique identifier of a conversation,

					references Conversations
date	DATE		NO	NONE	The date, time of the message
content	TEXT		NO	NONE	Content of the message

Table: PARTICIPANTS

Field Name	Data Type	Key	Null	Default	Description
email	TEXT	PK, FK(Users)	NO	NONE	User's email address as the unique identifier, references Users
conversation_id	INTEGER	PK, FK(Conversations)	NO	NONE	Unique identifier of a conversation, references Conversations

Table: BILLINGS

Field Name	Data Type	Key	Null	Default	Description
email	TEXT	PK, FK(Users)	NO	NONE	User's email address as the unique identifier, references Users
card#	INTEGER	PK	NO	NONE	Credit card number of user, unique identifier
card_type	TEXT		NO	NONE	Credit card type (MasterCard, VISA, American Express)
expiry_date	DATE		NO	NONE	Expiry date of credit card

Table: PAYMENTS

Field Name	Data Type	Key	Null	Default	Description
email	TEXT	PK, FK(Users)	NO	NONE	User's email address as the unique identifier, references Users
payment_id	INTEGER	PK	NO	NONE	Unique number that identify a payment
type	TEXT		NO	NONE	Payment type (ticket, upgrade)
payee	TEXT	FK(Billings)	NO	NONE	payee's email address, references Billings
card#	INTEGER	FK(Billings)	NO	NONE	Credit card number of payee, references Billings
amount	FLOAT		NO	NONE	Amount of a payment
payment_date	DATE		NO	NONE	Payment date

Table: PAPERS

Field Name	Data Type	Key	Null	Default	Description
paper_id	INTEGER	PK	NO	NONE	Unique number that identify a paper
title	TEXT	CK	NO	NONE	Title of paper, unique identifier with publish_date
publish_date	DATE	CK	NO	NONE	Publish date of paper, unique identifier with publish_date
latest_submit_date	DATE		NO	NONE	The date that this paper was submitted
url	TEXT		NO	NONE	url link of the paper

Table: PAPERS_TAG

Field Name	Data Type	Key	Null	Default	Description
paper_id	INTEGER	PK, FK(Papers)	NO	NONE	Unique number that identify a paper, references Papers
tag_name	TEXT	PK	NO	NONE	Name of paper tag

Table: PAPER_AUTHORED

Field Name	Data Type	Key	Null	Default	Description
email	TEXT	PK, FK(Users)	NO	NONE	User's email address as the unique identifier, references Users
paper_id	INTEGER	PK, FK(Papers)	NO	NONE	Unique number that identify a paper, references Papers

Table: PAPER REVIEWED

Field Name	Data Type	Key	Null	Default	Description
email	TEXT	PK, FK(Users)	NO	NONE	User's email address as the unique identifier, references Users
paper_id	INTEGER	PK, FK(Papers)	NO	NONE	Unique number that identify a paper, references Papers
event_id	INTEGER	PK, FK(Events)	NO	NONE	Unique identifier of an event, references Events
comment	TEXT		YES	NULL	Comment provided by reviewer
rate	INTEGER		YES	NULL	The rate give by reviewer after reviewed

flag	STRING		YES	NULL	To flag the review if there is problem
-------------	--------	--	-----	------	--

Table: EVENTS

Field Name	Data Type	Key	Null	Default	Description
event_id	INTEGER	PK	NO	NONE	Unique identifier of an event
name	TEXT		NO	NONE	Name of an event
type	TEXT		NO	NONE	Type of an event
from_date	DATE		NO	NONE	Start date of an event
to_date	DATE		NO	NONE	End date of an event
venue_id	INTEGER		YES	NULL	The id of venue which the event is held in, reference Venues
description	TEXT		NO	NONE	Description of an event
url	TEXT		NO	NONE	The url link of an event
poster_url	TEXT		YES	NULL	The poster url of an event
paper_deadline	DATE		YES	NULL	The final deadline of paper submission
review_tolerance	INTEGER		NO	DEFAULT AS 3	Tolerance for paper reviews
max_reviewer	INTEGER		YES	NULL	The maximum number of reviewers in an event
language	TEXT		NO	NONE	The language version of an event
reminder	TEXT		YES	NULL	The reminder for an event
privacy	ENUM		NO	“public” or “private”	The privacy of event, so creators can hide events
payee	TEXT	FK(billings)	NO	NONE	payee’s email address, references Billings
card#	INTEGER		NO	NONE	Credit card number of payee
contact_num	TEXT		YES	NULL	The contact number of security in an event

Table: EVENTS_TAG

Field Name	Data Type	Key	Null	Default	Description
event_id	INTEGER	PK, FK(Events)	NO	NONE	Unique identifier of an event, references Events
tag_name	TEXT	PK	NO	NONE	Name of event tag

Table: PAPER_SUBMITTED

Field Name	Data Type	Key	Null	Default	Description
paper_id	INTEGER	PK, FK(Papers)	NO	NONE	Unique number that identify a paper, references Papers
event_id	INTEGER	PK, FK(Events)	NO	NONE	Unique identifier of an event, references Events
status	TEXT		YES	NULL	The status of submitted paper (unreviewed, reviewing, reviewed)

Table: EVENT_ROLES

Field Name	Data Type	Key	Null	Default	Description
email	TEXT	PK, FK(Users)	NO	NONE	User's email address as the unique identifier, references Users
event_id	INTEGER	PK, FK(Events)	NO	NONE	Unique identifier of an event, references Events
role_name	TEXT		NO	NONE	The role of user participate an event

Table: EVENT_RATE

Field Name	Data Type	Key	Null	Default	Description
email	TEXT	PK, FK(Users)	NO	NONE	User's email address as the unique identifier, references Users
event_id	INTEGER	PK, FK(Events)	NO	NONE	Unique identifier of an event, references Events
rate	ENUM		YES	RANGE[1..5]	The rate of an event voting by attendee

Table: SESSIONS

Field Name	Data Type	Key	Null	Default	Description
event_id	INTEGER	PK, FK(Events)	NO	NONE	Unique identifier of an event, references Events
title	TEXT	PK	NO	NONE	Title of the session in an event, unique identifier
speaker_email	TEXT	FK(Users)	YES	NULL	Speaker's email address, references Users
start_time	DATE		NO	NONE	Start time of session
end_time	DATE		NO	NONE	End time of session

venue_id	INTEGER		YES	NULL	The id of venue which the event is held in, reference Rooms
room_name	TEXT		YES	NULL	The room name which the session is held in, reference Rooms
privacy	ENUM		NO	“public” or “private”	The privacy of event, so creators can hide sessions
conversation_id	INTEGER	FK(Conversations)	YES	NULL	Unique identifier of a conversation, references Conversations

Table: TICKETS

Field Name	Data Type	Key	Null	Default	Description
event_id	INTEGER	PK, FK(Sessions)	NO	NONE	Event id as the unique identifier, references Sessions
title	TEXT	PK, FK(Sessions)	NO	NONE	Title of the session in an event, unique identifier, references Sessions
name	TEXT	PK	NO	NONE	Name of ticket (early bird, normal, free, etc)
class	TEXT	PK	NO	NONE	Class of ticket (first, second, third)
type	TEXT	PK	NO	NONE	Type of ticket (normal, couple, hearing aid)
price	FLOAT		YES	NULL	Ticket price
description	TEXT		YES	NULL	Description of the ticket
start_date	DATE		NO	NONE	The start date of ticket sold
end_date	DATE		NO	NONE	The end date of ticket sold
quantity	INTEGER		NO	NONE	The quantity of this ticket type
num_purchased	INTEGER		NO	0	The number of ticket purchased
sale_end_date	DATE		YES	NULL	The due date of the ticket selling

Table: VENUES

Field Name	Data Type	Key	Null	Default	Description
venue_id	INTEGER	PK	NO	NONE	Unique identifier of a venue
name	TEXT		NO	NONE	Name of venue
type	TEXT		NO	NONE	Type of Venue (stadium, building)
street	TEXT		NO	NONE	Street name of venue
city	TEXT		NO	NONE	City name of venue
state	TEXT		NO	NONE	State of venue
country	TEXT		NO	NONE	County of venue
longitude	FLOAT		NO	NONE	Longitude of venue
latitude	FLOAT		NO	NONE	Latitude of venue

Table: ROOMS

Field Name	Data Type	Key	Null	Default	Description
venue_id	INTEGER	PK, FK(Venues)	NO	NONE	Unique identifier of a venue, references Venues
name	TEXT	PK	NO	NONE	Name of room, unique identifier
type	TEXT		NO	NONE	Type of room
capacity	INTEGER		NO	NONE	Capacity of room

Table: SEATS

Field Name	Data Type	Key	Null	Default	Description
venue_id	INTEGER	PK, FK(Rooms)	NO	NONE	Unique identifier of a venue, references Rooms
name	TEXT	PK, FK(Rooms)	NO	NONE	Name of room, unique identifier, references Rooms
seat_num	INTEGER	PK	NO	NONE	Seat number of the room, unique identifier of seat

Table: TICKET_RECORD

Field Name	Data Type	Key	Null	Default	Description
record_id	INTEGER	PK	NO	NONE	Record id as the unique identifier (the reason create this is all because MySQL database can't take too much attribute as primary)

event_id	INTEGER	PK, FK(Tickets)	NO	NONE	Event id as the unique identifier, references Ticket
title	TEXT	PK, FK(Tickets)	NO	NONE	Title of the session in an event, unique identifier, references Ticket
ticket_name	TEXT	PK, FK(Tickets)	NO	NONE	Name of ticket (early bird, normal, free, etc), references Ticket
class	TEXT	PK, FK(Tickets)	NO	NONE	Class of ticket (first, second, third), references Ticket
type	TEXT	PK, FK(Tickets)	NO	NONE	Type of ticket (normal, couple, hearing aid), references Ticket
venue_id	INTEGER	FK(Rooms)	YES	NULL	Unique identifier of a venue, references Room
room_name	TEXT	FK(Rooms)	YES	NULL	Name of room, unique identifier, references Room
seat_num	INTEGER		YES	NULL	Seat number of the room, unique identifier of seat
email	TEXT	FK(Users)	YES	NULL	Attendee's email address , references User

Table: RESOURCES

Field Name	Data Type	Key	Null	Default	Description
venue_id	INTEGER	PK, FK(Rooms)	NO	NONE	Unique identifier of a venue, references Rooms
room_name	TEXT	PK, FK(Rooms)	NO	NONE	Name of room, unique identifier, references Rooms
name	TEXT	PK	NO	NONE	Name of resource, unique identifier
type	TEXT		NO	NONE	Type of resource
number	INTEGER		NO	NONE	Number of resource

Third-party Libraries

- Laravel – Based on PHP, Laravel is a web application framework that is based on the Model-View-Controller architectural pattern. It simplifies functions such as database access and query building. Laravel is a popular PHP framework which ensures that help resources are available online.

Reference: <http://laravel.com/>

- PHPUnit – A unit testing framework designed for testing PHP scripts and classes. We chose this testing framework because has built-in support with Laravel which makes it an ideal choice for testing our PHP scripts.

Reference: <https://phpunit.de/>; <http://laravel.com/docs/5.1/testing>

- jQuery – A JavaScript library that simplifies client-side JavaScript programming. jQuery has many in-built features, including shorthand Ajax requests to simple animations, that reduce programming time significantly.

Reference: <https://jquery.com/>

- Bootstrap – An open-source front-end framework that is designed to build websites and web-applications quickly. It integrates HTML, CSS and JavaScript into one coherent framework that is supported the major web browsers. It also has good support for websites built for mobile devices due to its responsive web design.

Reference: <http://getbootstrap.com/>

- Amazon Web Services (AWS) – AWS provides many different web services that range from networking to database services. The Event Organizer is hosted on Amazon EC2 with the MySQL database on Amazon RDS. This is supported by Amazon S3 which is the storage services of AWS.

Reference: <https://aws.amazon.com/>

- Mailchimp – An email service that Eventure will use to send confirmation emails to users and reminder emails to event attendees.

Reference: <http://mailchimp.com/>

- iRedMail – An open-source mail server software that will be used as the mail server of Eventure. It will be installed on the AWS EC2 web service.

Reference: <http://www.iredmail.org/>

- CocoaPods – This technology is a library of Objective-C libraries. It acts as a dependency manager for both Swift and Objective-C applications. This facilitates code reuse and reduces the amount of time spent in writing and testing code.

Reference: <https://cocoapods.org/>

- Alamofire – a wrapper for the default iOS HTTP networking that makes it more elegant

Reference: <https://github.com/Alamofire/Alamofire>

- SwiftyJSON – a wrapper the default iOS JSON interpretation that makes it more elegant

Reference: <https://github.com/SwiftyJSON/SwiftyJSON>

- CryptoSwift - framework for hashing - this was used to hash the user password on device so that the application has both client and server side hashing

Reference: <https://github.com/krzyzanowskim/CryptoSwift>

Testing Plan

Testing is generally considered a part of the development lifecycle that is performed at a late stage but in reality it is a phase which can start at the very beginning of the SDLC. As we are not using a Test Driven Development Methodology we will not be performing testing before we write code. We are using a RaPID Development Methodology (rapid prototyping and iterative development) therefore we will perform testing at the end of each iteration.

Testing inaccuracy

All testers either have optimistic inaccuracy or pessimistic inaccuracy. For the scope of this project in the limited time frame we have and the large scale, we will be inclined to prefer an optimistic inaccuracy approach. Pessimistic inaccuracy is when developers cannot prove that the software achieves 100% correctness therefore they don't accept the software. Optimistic inaccuracy is when developers cannot detect any faults through thorough testing and because of this they are happy even though they cannot prove 100% correctness. This is generally the case with most systems that are not safety critical like a conference/event management system.

Techniques

It is prevalent that not one single technique for testing is optimal. A combination of testing techniques increases testing effectiveness. Each testing technique has a different purpose and tradeoffs.

Acceptance testing will be done toward the mid-late stages of development. This will be done by doing alpha testing. Alpha acceptance testing is when the company acquires individuals to use the product under a controlled environment.

When **alpha testing** is done, the developer runs the test by having an individual who is likely to use the system. We introduce them to the product and lay out a bunch of functionalities that the system can perform. For example, book a ticket to "UOW The Movie" with the user's preferences. Rather than telling the individual to click this a particular button and induct them through the process by telling them the goal they wish to achieve we can observe their behaviour and their interaction with the interface. If all goes to plan the interface should be intuitive (speak for itself). If the user appears confused or takes too long to perform the task within the limits set by the non-functional requirements, improvements should be made to the system, given the project is running smoothly and we are able to complete the project in time. In outlining this it is important to note that this project/assignment has a fixed due date unlike real life projects, which although they have a due date, are generally flexible to allow small desirable items to be implemented/checked/testing, etc. Clients are generally willing to wait extra time in order to improve aspects such as usability. As we are using RaPID we will be performing rapid prototyping of the user interface. It will not be necessary to perform alpha testing on every single rapid prototype (assuming that part of the code is function). We feel it is best to select one stage when we will perform alpha testing and based on feedback we will perform more changes to the GUI (continue rapid prototyping of GUI) until we have satisfied all issues detected. We should then run alpha testing on the new build with the existing testers which had issues using the software and a new group of individuals.

Future development

The system we are designing is not designed solely to achieve 100% functionality. The requirements have been designed so that all Core, Main and possibly some Stretch requirements are achievable within the time given, team experience and available equipment. The Stretch goals and Dream goals have been implemented so that if the system can be further developed after the project has been completed. If further perfective maintenance or system modifications are performed, further tests should be executed to ensure the quality of the system.

Sensitivity

We aim to have all testing at a high degree of sensitivity. Sensitivity in testing is an Analysis and Testing Principle that ensures that the outcome of a test is consistent. This is crucial for us as our development team does not want to have test cases that passes a majority of the time but fails only once. Assuming the same input (a test case), the output should be the same each time (a pass or fail of the test case or the same data output/effect).

Assertion

For the web end we are using PHP which come with the ability to use assertions. This will become useful when we perform testing. It is important to note that there is different syntax for different versions of PHP so all developers must be using the same version of PHP.

For iOS we can also use assertions to test the mobile application. iOS has a NSAssertionHandler object which can be used for testing.

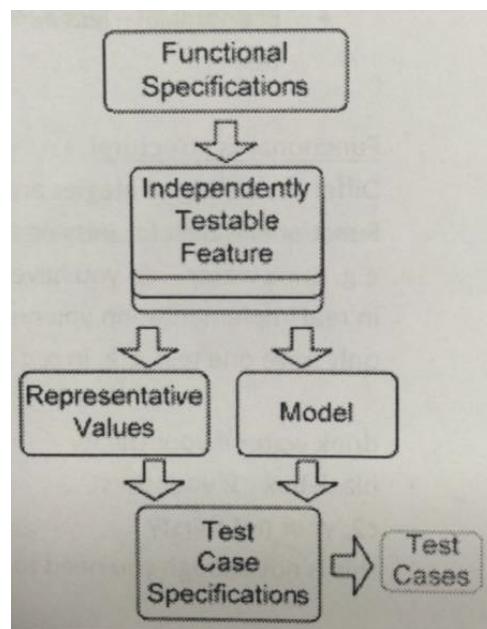
Test cases

The test case specification will be developed at the end of the first iteration.

A test case specification must be satisfied by at least one test case.

Each test case must have specified input, execution conditions and pass/fail criterion. They should be outlined if they are for the Website or for the iOS app.

The diagram below shows the process of how test cases can be formed from requirements.



Black Box and White Box Testing

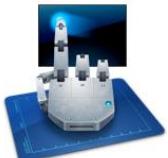
Black box testing, also known as functional testing, is testing in comparison to the requirements (SRS). White box testing, also known as structural testing, is the testing of the actual code itself.

The methods of performing black box testing are unit, integration, system and regression testing. We should aim to perform all these types of testing on core parts of the system. The methods for performing white box testing are unit testing and integration testing which we should also aim to cover.

Covering both black box and white box testing is optimal and the best way to achieve black box and white box testing is to do black box runtime monitoring first of all. Once this is done, the covered branches must be identified and test cases should be created for these branches.

Testing tools

- Jenkins
Jenkins is a tool which we can use for testing. It was originally developed for testing web apps but since has become the industry standard for open source iOS testing. It is therefore suitable for our project as we can use it for both the website and the iOS application.
- Xcode for iOS
XCode has built in code coverage tools. static analyser and a linter tool. Linting is the process of running a program that will analyse code for potential errors. It should enforce styles and conventions for the chosen language.
- SwiftLint is one linting tool that can be used for swift
- XCTest framework
UI testing framework for iOS apps. It also allows us to take screenshots at each stage throughout the application
- Unit Testing
XCode has built in unit testing. Examples and guides can be found on iosunittesting.com. We should be able to perform unit testing on both Objective C and Swift code at each iteration.
- UIAutomation
UIAutomation is a tool within XCode's instruments. We have found a useful guide to beginning automated testing in XCode. Another tool we could use is Frank which allows structured text test such as acceptance/requirements tests to be executed against the iOS application.
- Core Animation
Measures application graphics performance as well as CPU usage of a process via time profiling.
- Time Profiler
Performs low-overhead time-based sampling of processing running on the system's CPU.
- Leaks
Measures general memory usage, checks for leaked memory and provides statistics on object allocations by class as well as memory address histories for all active allocations and leak blocks.



Automation



Core Animation



Time Profiler



Leaks

- Energy Diagnostics
Diagnostics regarding energy use as well as basic on/off state of major device components.
- Core Data
Traces Core Data file system activity including fetches, cache misses and saves.
- Code Coverage
XCode had the ability to gather code coverage data which the selection of a check box. Other possibilities are GCOV, Slather and coveralls.io. coveralls is a web service that can be used for C/C++, D, GO, Haskell, JAVA, Javascript and node, Julia, LUA, .NET, PHP, Python, R, Ruby/Rails, Scala, Objective C and OCAML for code coverage. It would therefore be suitable to test the code coverage of PHP and Objective C code. This is free for open source projects on Github or BitBucket. For swift code coverage SwiftCov is a possible tool.
- Payment Research
Based on research we have conducted it is seem that collis test tool is the industry standard for mobile payment testing. This would be good but most likely unattainable as it is paid testing software which is most likely expensive.



Energy Diagnostics

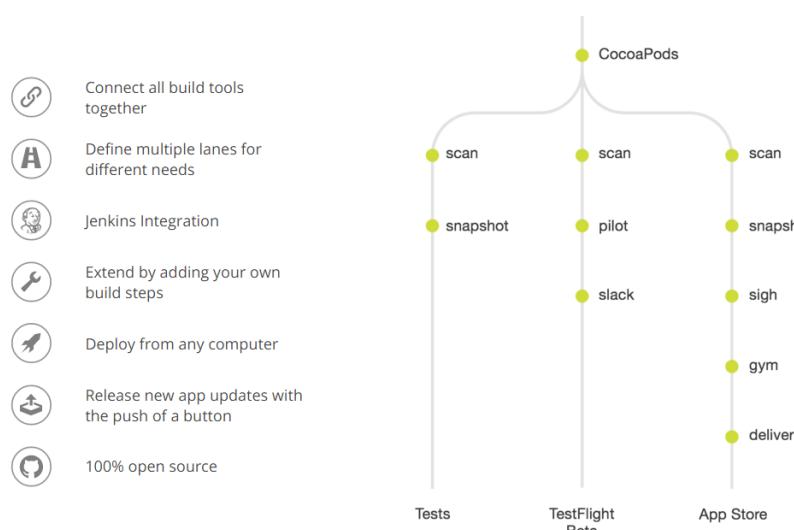


Core Data

We have found this link on the PayPal developer sight which uses V.Zero which accepts, PayPal, credit card, apple pay, android pay, bitcoin and venmo payments. They have an PayPal SDK and a library for iOS called Braintree.

- Fastlane

Fastlane can be useful during release. It is a tool that allows deployment pipelines to be run on different environments that helps unify and automate the process. It has integration with CocoaPods and Jenkins and is open source. A list of tools Fastlane provides is shown below.



Meet the fastlane family (iOS)



deliver

Upload screenshots, metadata and your app to the App Store using a single command

Star 2,216



snapshot

Automate taking localized screenshots of your iOS app on every device

Star 1,630



frameit

Quickly put your screenshots into the right device frames

Star 848



PEM

Automatically generate and renew your push notification profiles

Star 488



sigh

Because you would rather spend your time building stuff than fighting provisioning

Star 848



produce

Create new iOS apps on iTunes Connect and Developer Portal using the command line

Star 132



cert

Automatically create and maintain iOS code signing certificates

Star 128



gym

Building your iOS app has never been easier

Star 491



scan

The easiest way to run tests of your iOS and Mac app

Star 65

Web Testing

Number	Name	Whether successful
M_1	Log in button	Y
M_2	Sign up button	Y
M_3	Search button	Y
M_4	Help button	N
M_11	Log out button	Y
M_12	Show more detail	Y
M_13	More detail button	Y
M_14	Event button	N
M_15	Other website	Y
M_16	About us button	N
M_17	Feedback button	N
M_18	Membership button	Y
M_19	Site map button	N
M_20	Connect with us	N
M_21	My account button	Y
L1	Input email	Y
L2	Input password	Y
L3	Submit button	Y
S_1	Input email	Y
S_2	Input user name	Y
S_3	Password	Y
S_4	Confirm password	Y
S_5	Submit button	Y
A_1	My dashboard	Y
A_2	My profile button	Y

A_3	Events attending	Y
A_4	Event manage	Y
A_5	Upgrade account	Y
A_6	Create event	Y
A_7	My papers button	Y
Function test		
TEST1	Log in	Y
TEST2	Log out	Y
TEST3	Sign up	Y
TEST4	Change profile	N
TEST5	Create event	Y
TEST6	Upload paper	Y
TEST7	Create room	Y
TEST8	Create session	Y
TEST9	Create ticket	Y
TEST10	Modify event detail	Y
Test 11	Add expertise	Y
Test 12	Purchase ticket	N

Main page

M_1

Function name: log in button

Function in display:

Log In

Function result: enter log in page

Correct operation response: press it

Login

Email Address

Password

SUBMIT ➤

Error operation response: no need

Testing result/comment: Testing successful

M_2

Function name: Sign up button

Function in display:

Sign Up

Function result: enter the sign up page

Correct operation response:

Sign up as a member of Conf+

Email Address

Username

Password Confirm password

SUBMIT ➤

Error operation response: no need

Testing result/comment: Testing successful

M_3

Function name: search button

Function in display:



Function result: start search function

Correct operation response: nothing happened

*Error operation response: no need
Testing result/comment: successful*

M_4

Function name: Help button

Function in display:

Help

Function result: start help function

Correct operation response: nothing happened

Error operation response: no need

Testing result/comment: Testing fail

Need have function

M_11

Function name: log out button

Log Out

Function in display:

Function result: log out the account

Correct operation response: change to log in button

Error operation response: no need

Testing result/comment: success

M_12

Function name: show more detail button

:

Function in display:

Function result: show more detail on photo

Event	
Name	X
Event	
Description	

Correct operation response:

Error operation response: no need

Testing result/comment: successful

M_13

Function name: more detail button

More details

Function in display:

Function result: enter this event page

Correct operation response:

BUY TICKET	
Start Date - End Date	2016-06-06 12:00:00 - 2016-06-08 20:00:00
Location	Invalid, Venue, ID
Description	Event Description
Tags	No Tags X

*Error operation response: no need
Testing result/comment: successful*

M_14

Function name: enter all event page button



Function in display:

Function result: enter all event page button

Correct operation response: nothing happened

Error operation response: not need

Testing result/comment: unsuccessful testing

M_15

Function name: other website buttons



Function in display:

Function result: enter other website

Correct operation response: enter

Error operation response: no need

Testing result/comment: success

M_16

Function name: about us button

About us

Function in display:

Function result: enter about us page

Correct operation response: nothing happened

Error operation response: no need

Testing result/comment: unsuccessful

M_17

Function name: feedback for us button

 Feedback

Function in display:

Function result: enter feedback page

Correct operation response: nothing

Error operation response: not need

Testing result/comment: unsuccessful

M_18

Function name: enter his account

 Membership

Function in display:

Function result: enter log in page

Correct operation response: enter log in page

Error operation response: not need

Testing result/comment: successful

M_19

Function name:

 Site Map

Function in display:

Function result:

Correct operation response:

Error operation response:

Testing result/comment:

M_20

Function name: connect us button

 Connect with us

Function result: see all in connect info

Correct operation response: nothing

Error operation response: not need

Testing result/comment: unsuccessful

M_21

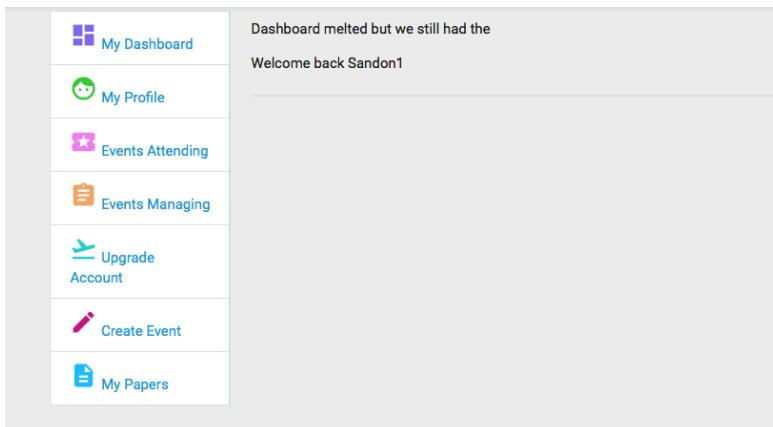
Function name: my account button

 My Account

Function in display:

Function result: enter my account page

Correct operation response:



Error operation response: not need

Testing result/comment: success

Log in page

L_1

Function name: input the email address

Function in display:

Function result: input the email address

Correct operation response: system can get email

Error operation response: system rejects the input

Testing result/comment: success

L_2

Function name: input the password

Function in display:

Function result: put the input to database

Correct operation response: enter next page

Error operation response: can't log in

Testing result/comment: success

L_3

Function name: submit button in log page

Function in display:

Function result: enter my account page

Correct operation response: change the log in button to my account and skip to main page

Error operation response: can't work

Testing result/comment: success

Sign up page

S_1

Function name: input email



Email Address

Function in display:

Function result: put the email to database

Correct operation response: save email

Error operation response: reject the input

Testing result/comment: success

S_2

Function name: input user name



Username

Function in display:

Function result: put the use name to database

Correct operation response: save username to database

Error operation response: can't sign up

Testing result/comment: success

S_3

Function name: input password



Password

Function in display:

Function result: put the password to database

Correct operation response: can sign up successful

Error operation response: can't sign up

Testing result/comment: success

S_4

Function name: confirm password



Confirm password

Function in display:

Function result: compare password and ensure they are same

Correct operation response: can sign up successful

Error operation response: output the error message

Testing result/comment: success

S_5

Function name: submit button

SUBMIT ➤

Function in display:

Function result: make sure everything is correct

Correct operation response: sign up successful and enter log in page

Error operation response: can't work and output error

Testing result/comment: success

Account page

A_1

Function name: my dashboard button



Function in display:

Function result: enter main page of account

 My Dashboard My Profile Events Attending Events Managing Upgrade Account Create Event My Papers	Dashboard melted but we still had the Welcome back Sandon1
--	---

Correct operation response:

Error operation response: no need

Testing result/comment: success

A_2

Function name: my profile button



Function in display:

Function result: enter my profile page

Correct operation response:

Change your settings

 fan@uawic.com

 fanusername

 Password.....  Confirm password

 Title  First Name  Last Name

 _____

 Street address

Error operation response: NO NEED

Testing result/comment: success

A_3

Function name: events attending button



Function in display:

Function result: view attending event

Correct operation response: error(back to main page)

Error operation response: not need

Testing result/comment: unsuccessful

A_4

Function name: events managing button



Function in display:

Function result: enter event-managing page

Correct operation response:

Events Managing

Name	Type	Start Date	End Date	Actions

Error operation response: no need

Testing result/comment: unsuccessful//because we still can use this button when we don't have upgrade account

A_5

Function name: upgrade account button



Function in display:

Function result: enter upgrade page

Play your part Upgrade your Conf+ account

What can you do with an upgraded account?

- ✓ Create event
- ✓ Edit event
- ✓ Submit paper
- ✓ Review paper
- ✓ Assign paper

UPGRADE NOW FOR ONLY \$4.99!

Correct operation response:

Error operation response: no need

Testing result/comment: success

A_6

Function name: create event button



Function in display:

Function result: enter create event page

Correct operation response:

Create Event

	Event Name
<hr/>	
	Event Description
<hr/>	
	Start Time
	End Time
<hr/>	
	Venue
<hr/>	
<input type="radio"/>	Create New Venue
<input type="radio"/>	Use Preexisting Venue
<hr/>	
	Event type
<hr/>	
<input type="radio"/>	Event

Error operation response: NO NEED

Testing result/comment: SUCCESS

A_7

Function name: my papers button



Function in display:

Function result: enter paper manage page

Correct operation response:

My Papers

[Upload New Paper](#)

Title	Publish Date	Last Submit Date	Manage
<hr/>			

Error operation response: no need

Testing result/comment: success

Test 1

Name: log in

- Use Case: 1. Press the log in button
 2. Input the email and password
 3. Press the submit

Correct Outcome: the sign up button change to my account button and the log in button change to log out button

Actual Outcome: the sign up button change to my account button and the log in button change to log out button

Extra Information:

Test 2

Name: log out

- Use Case: 1. Press the log out button

Correct Outcome: the log out button change to log in button, the my account button change to sign up button

Actual Outcome: same

Extra Information:

Test 3

Name: sign up

- Use Case: 1. Press the sign up button
 2. Input the user name, email address, password, confirm password
 3. Press submit button

Correct Outcome: skip to log in page and can use new account to log in

Actual Outcome: same

Extra Information:

Test 4

Name: change profile setting

- Use Case: 1. Press the log in button

- 2. Input email and password, then press the submit
- 3. Press the my profile
- 4. Input every information in every input space
- 5. Press submit button

Correct Outcome: Every information will be change and save

Actual Outcome: same, but information cannot save when he log out.

Extra Information:

- 1. Using postman can post and get the data

Test 5

Name: create event

- Use Case: 1. Log in account
 2. Press create event

3. Input all information

Create Event

Event Name

Event Description

Start Time

End Time

Venue
 Create New Venue
 Use Preexisting Venue

Country sadsa	State asda	City sads
------------------	---------------	--------------

sadsd, sadasd

4. Press submit

Correct Outcome: the event can be created

Actual Outcome: same (~~just use preexisting venue can create~~)

~~Extra Information: if use create new venue, the create event will not work.~~

~~FIXED 1/6/2016~~

Test 6

Name: upload paper

Use Case: 1. Log in a upgraded account

2. Press the my paper button

3. Press the upload new paper button

4. Input every detail about paper, then press submit

Upload New Paper

	Paper Title test11		Published Date 2015-12-12 12:22:22
	aaa.png		
SUBMIT PAPER			
Submitting...			

Correct Outcome: the paper will be showed in my paper

Actual Outcome: same like correct outcome

Extra Information:

Test 7

Name: create a room

Use Case: 1. Log in an upgraded account

2. Press event managing
3. Press a “go to event dash”
4. Press venues

Correct Outcome: the new room will appear in the bottom.

Actual Outcome:

Rooms

Rooms			
BACK ←			
Room Added			
Name	Type	Capacity	Action
gogo	room	100	ADD

Extra Information: ~~the hint information showed room added, but it is not show the room detail in somewhere, so we need fix it.~~

Fixed, all good

Test 8

Name: create session

Use Case: 1. Log in an upgraded account

2. Press event managing
3. Press a “go to event dash”
4. Press session
5. Input all detail. (Room name should exist room.)
6. Press add

Correct Outcome: the session will show at the bottom.

Actual Outcome: same

Sessions				
Title	Start Time	End Time	Room Name	Action
				ADD
All days	2015-10-12 12:22:22	2016-10-12 12:22:22	Room 1	EDIT DELETE
half day	2015-10-12 12:22:22	2016-09-12 12:22:22	gogo	EDIT DELETE

Extra Information: it is working but this session information is not show in event dash, need fix.

Test 9

Name: create a ticket

Use Case: 1. Log in an upgraded account

2. Press event managing
3. Press a “go to event dash”
4. Press tickets
5. Input all detail then press create

Correct Outcome: the new ticket will appear at bottom

Actual Outcome: same as wish

Session	Name	Class	Type	Price	Description	Start Time - End Time	Purchased / Total	Action
All days	asdad	12	12	\$999.99	12dadasdsadadada	2015-10-12 - 2016-10-12 12:22:22	0 / 111	<button>Delete</button>
half day	adult	a	d	\$200.00	for adult	2015-11-12 - 2015-11-22 12:22:22	0 / 200	<button>Delete</button>

Extra Information: this function is working; only thing is the new ticket is not shown in event dash, need fix

Test 10

Name: modify event detail

Use Case: 1. Log in an upgraded account

2. Press event managing
3. Press a “go to event dash”
4. Press basic details
5. Change some detail about event
(Menu can let user change different information)

Correct Outcome: the detail will be saved

Actual Outcome: same as wish

Extra Information:-poster upload is not working

1. But no notification to remain user, the poster is added
2. Don't show original data in event detail page, need add.

Test 11

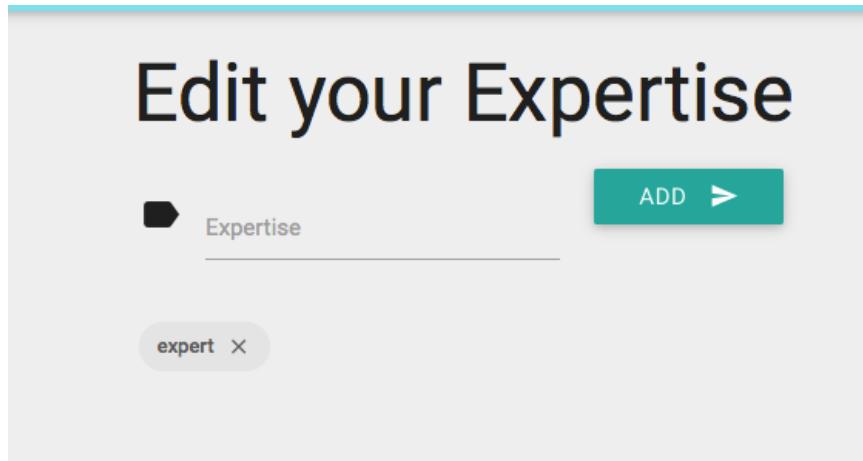
Name: add expertise

Use Case: 1. Log in an upgraded account

2. Press expertise
3. Input tag and press add

Correct Outcome: the tag will show

Actual Outcome: same as wish



Extra Information:

Test 12

Name: purchase ticket

Use Case: 1. Log in an upgraded account

2. Press an event
3. Press a “buy ticket”
4. Choose a ticket and change the number
5. Press next
6. Input the name and email and press the purchase

Correct Outcome: we can buy the ticket

Actual Outcome: when we press the purchase, nothing happen

Extra Information:

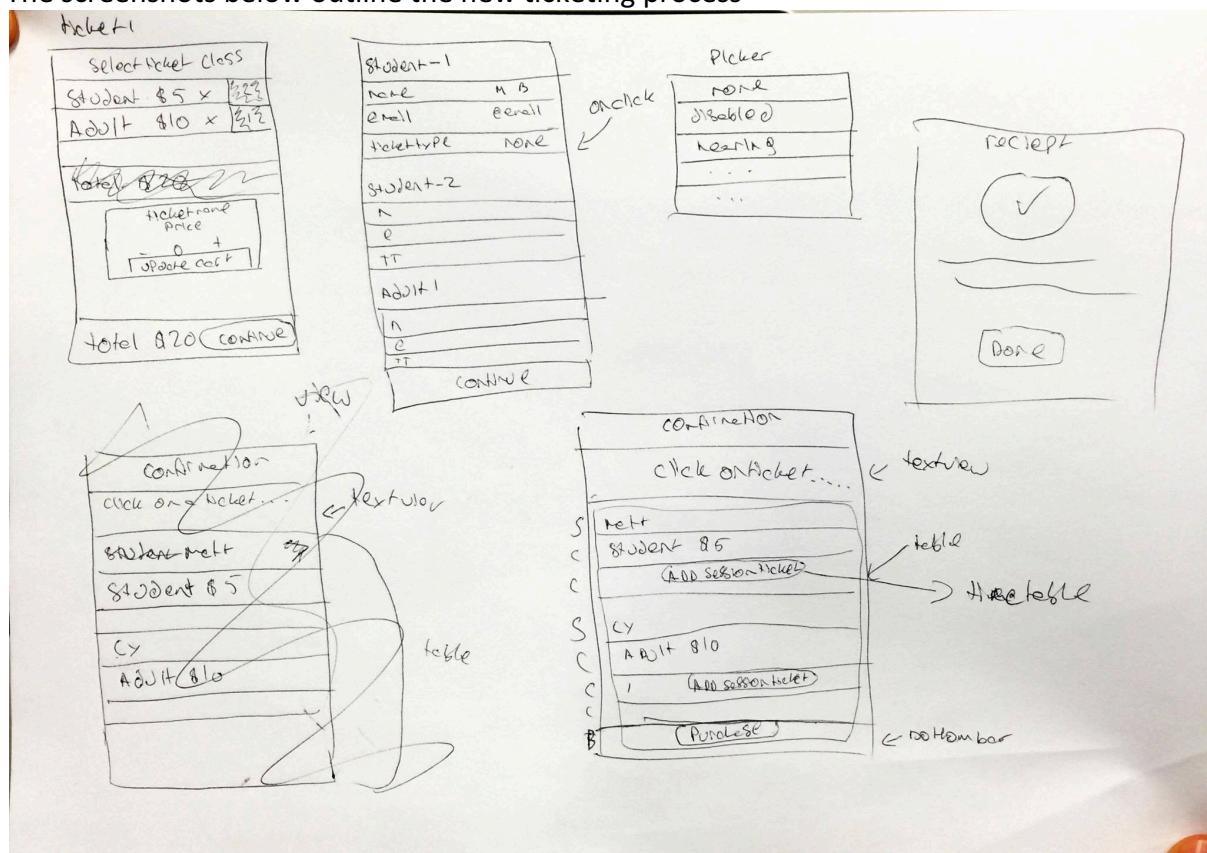
iOS Testing

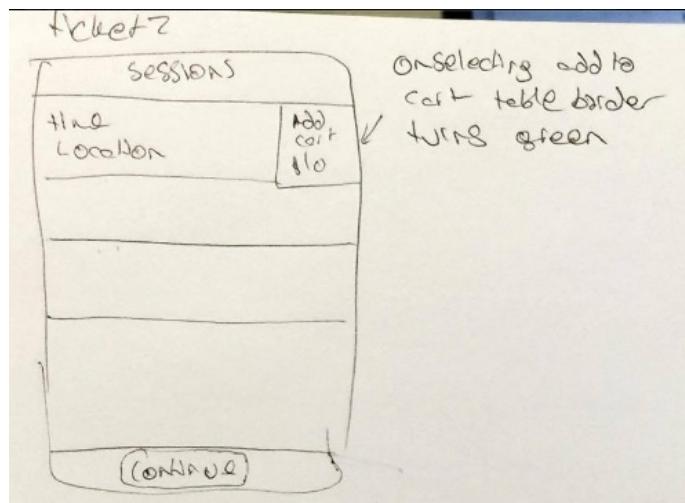
As part of testing, the iOS team performed user testing on an experienced mobile developer. The mobile developer was given some context of the application and the testers observed his behaviour. During the test the mobile developer gave real-time feedback on what he was thinking and expecting on certain screens. This gave the testers valuable feedback and improved the usability of the app extensively.

Some key points to make that were improved was that the entire ticketing process was quite confusing. Originally the ticketing process had a branch where the user could select a event ticket and then either continue to buy session tickets or click buy to purchase the entry tickets alone. The concept of having continue and buy as two buttons and a branching process flow was quite confusing. To resolve this, the iOS team did a new prototype of the ticketing process which took into consideration the feedback provided. The new process flow has the ability to first purchase tickets for the event then you are taken to a screen to purchase session tickets if the user wants to purchase session tickets they can do so, if not they can continue to payment.

Another important point discovered during user testing is that, although it is not necessarily bad practice or not "good UI", the mobile developer didn't like the idea of clicking on an event and seeing a dashboard without any information. he believed that there should be some information displayed upon clicking on the "my events" tab

The screenshots below outline the new ticketing process





My Events

Present	Past
---------	------

(Mo)

Event Name
Date to - Date from
city, country

C Back

Timetable

Dynamic

If admin show admin cell

Event Details

(Mo)

Date Event name

Date

Buy

desc

map

location

Timetable
Attendance
Invitations
Message
tickets
Admin