Team CASA

15/03/2019

Team Project - Project Plan

Team Members:

Name Student Number Primary Role Secondary Role

Cian Twyford - C16380866 - Code Developer - Planner
Amy McNabb - C16323781 - Team Leader - Developer
Adam Ryan - C16302271 - Tester - Censor
Stephen Alger - C16377163 - Designer - Developer

Roles:

- *Team lead – Amy McNabb*
- *Developer - Amy McNabb, Stephen Alger, Cian Twyford*
- *Tester – Adam Ryan*
- *Documentation – Amy McNabb, Stephen Alger, Cian Twyford*

Project Title

CommunityLife Application

Version Number:1

Change History:

Date	Author	Comments
11/03/2019	Cian Twyford	Initial Start on document & Proposed Approach
14-03-2019	Amy McNabb	Roles Added, Original Timeline of tasks
15/03/19	Stephen Alger	Summary added, Technical Requirements added, Deliverables, Task breakdown edited.

Summary

This project involves the undertaking of creating and delivering a platform which will promote events which are accessible and welcoming to people with special needs. Initially this project was undertaken with the belief that it should be delivered as a website to replace the current CommunityLife.ie setup, however through deep thought and consideration of client feedback, we believe it would be unfeasible to rework the current system. The recurring theme when communicating with the prospective end users was that a mobile application delivery would be far more effective in succeeding. Therefore as a team we have made the decision to design for Android and IOS for both Mobile and Tablet devices as the overwhelming majority of Clients had direct access to a tablet or smartphone.

Proposed Approach

0. Overall approach

We shall be using Test Driven Development
Risks associated with that Test Driven Development:
Risks: it slows down development for rapidly iterative start up environments, the implementation code may not be ready for some time due to writing tests first. and how you will deal with the risks. However, this will pay off in the long run. Creating tests for failure also be tedious, but it also pays off in the end.

1. Requirements Phase:

Any additional requirements will be gathered when meeting with the clients. The risks here can be that we learn of a requirement at a late stage. We will solve this by dividing the workload based on team members tasks for that week and incorporate it into the week's plan.

2. Design Phase:

The Principles of Universal Design. Principles 1 -6 will be used.
A huge risk is are clients not being able to operate the app in a hassle-free manner across the board for our entire user base. We shall resolve this by conducting behavioural and interaction studies with the users each week taking into account methodologies such as "think out loud" etc.

3. Implementation Phase:

We will be using Java on Android Studio, on Windows. We will use Android version 6.0

Team Project - Project Plan

Marshmallow to comply with a large amount of devices. A risk may be that this version is not agreeable with some newer functions. This can be easily resolved by implemented the function version and synching it.

4. Testing Phase:

As said in the Overall Approach, we shall be using Test Driven Development. We will also use JUnit. Unit testing folders with example code will be made. We will test how the app is going with the clients too. A risk of unit testing is that it does not show the absence of errors. To resolve this, we must run every test case and scenario.

Deliverables

Code	Name	Priority	Description
1	*Eventrbrite API interaction*	*Base*	**Create HTTP requests using API* **Send request to Eventrbite servers* **Parse JSON response* **Convert into custom ListView elements*
2	Database Implementation		Firebase & Local Device DB implementation
3	Base UI Package		Home UI, Categorisation of Events, Listview of Events, Ability to favourite event type & individual event
4	Core Features		Calendar, Notification and GPS functionality added. Along with link to webform to allow specific end users to promote an event not current on the system.

Technical Requirements

- *Android Studio for Android.*
- *Xcode for iOS Development.*
- *Eventbrite API - The Eventbrite API is REST-based, uses OAuth2 for authentication (straightforward to use) , and returns responses in JSON. Interaction with this API will require HTTP requests and a JSON parser.*
- *Create in-house event system which allows for manual event uploads, eventbrite will provide only a small subset of the total events promoted on the system - therefore we need a server hosting to respond to requests.*
- *Database Architecture: Firebase for Android & iOS - track minimal amount of user data to personalise experience.*
- *Extensive Cross Platform Testing on a multitude of device configurations.*

Team Project - Project Plan

Related Documents

Document Title	Author(s)	Description
Week 7 - Project Plan Presentation	Amy McNabb, Cian Twyford, Stephen Alger, Adam Ryan	Risks, Risk Management & Mitigation and Timeline documented. Proposed project approach and technical requirements further elaborated upon since the Design Document. Member Roles are more precise since initial formation.
Week 5 - Design Presentation	Cian Twyford, Stephen Alger, Amy McNabb, Adam Ryan	Design defence delivered to the panel, clearly putting forward our systems design, including the appropriate UML, screenflow and summarising the proposed system architecture.
Screenflows PDF	Stephen Alger, Amy McNabb, Cian Twyford	A collection of Screenflows showing a clear progression from iteration zero to the latest, adopting feedback from the client.
Week 5 - Design Document	Amy McNabb, Stephen Alger, Cian Twyford	Design outline from the team showing core features of the app and proposed presentation.

Project Plan

Week 7:

Phase	Task	Priority	Name	Description
0	Project Plan	high	Amy, Adam	Intro, Time, Work Allocation, Milestone, Summary, Risk slides
0	Project Plan	high	Cian	Critical Path slide
0	Project Plan	high	Stephen	Sequence of activities slide

Week 8:

Phase	Task	Priority	Name	Description
1	Retrieve JSON data (Android)	#1	Amy	Connect with the Eventbrite API using OKhttp or Volley and collect and parse the necessary JSON data.
1	Retrieve JSON data (iOS)	#1	Stephen	Connect with the Eventbrite API and parse the necessary JSON data. (iOS Implementation)
1	Testing	#2	ALL	Testing API interaction
1	Firebase Database Design	#3	Stephen	Hosted Database design which will act At backend of system.
1	Local Device DB Design (Android &	#3	Stephen & Cian	Device storage – for offline & experience personalisation.

Team Project - Project Plan

	iOS)			
1	Testing	#4	ALL	Testing Database Design & Implementation

Week 9:

Phase	Task	Priority	Name	Description
2	Make RecyclerView list	High	Cian, Amy & Stephen	Make a RecyclerView list which will contain the JSON data in an orderly fashion. Make each item on list clickable. Implementing a custom list with custom list adapter.
2	Home UI with Event Categorisation	High	Cian,Amy & Stephen	UI Design implementation in coherence with latest screenflow specification, Implementing event categorisation through visuals.
2	Login Functionality	Medium	Cian,Amy & Stephen	The implementation login (QR).
	Favourite Functionality	Medium	Cian, Amy & Stephen	The ability to add an event type to favourites or a specific event.
2	Testing	High	Adam Ryan	Testing this weeks features thoroughly

Week 10:

Phase	Task	Priority	Name	Description
3	Favourite Event ListView	High	Cian, Amy & Stephen	Display favourited event types and favourited individual events which the client wishes to attend.
3	In-House Event Creation	High	Cian & Stephen	Creating the point of event creation for event promoters and community leaders. I.E. events that are not being pulled from Eventrbite, perhaps can be achieved through a web form based event registration.
3	GPS Integration	Medium	Amy & Stephen	Find activities based on users location, perhaps activities within 30KM
3	Calendar Integration	Medium	Cian & Stephen	Events which have been favourited automatically get added to client's calendar with a reminder pre-set.
3	Testing	High	Adam Ryan	Testing this weeks features thoroughly

Week 11:

Phase	Task	Priority	Name	Description
4	UI Polish	High	ALL	Final UX is Uniform cross platform
4	Final Testing	High	Adam Ryan	Make sure that navigation between pages are functionable and final testing of app's code.
4	Implement Text to speech	High	Amy, Cian Stephen	Give users the option to have the text on the screen read out for them.
4	Final Task (woop)!	High	Adam, Amy, Cian, Stephen	Documentation, submission, presentation and demonstration of app.

Team Project - Project Plan

Risk Register				
Risk ID	Risk Title	Probability [Hi,Med,Low]	Impact	Action
1	Getting late requirements from clients	Medium	More work!	Separate work evenly between team members based on their tasks priority and plan to avoid overload
2	Clients not understanding apps design or functionality	High	Clients not being able to use the app	Let clients test what we have done so far and see how their interaction with the app is
3	Android does not recognise newer functions late requirements from client	Low		Find alternative method, possibly resolvable by implementing the function and the version and synching it.
4	Unit testing not showing absence of errors	High	We may think the code/functionality is correct when it actually is not	Run every test case and scenario
5	System Does not provide enough relevant events to user	High	Application does not get widely adopted & does not solve the original problem for clients	Provide multiple methods of event addition to system. I.E. take in Eventbrite events and let community groups or event promoters add their own.

Team Project - Project Plan