

# GigMeln Project Topic Brief

---



<b>Introduction</b>	<b>4</b>
Overview	4
<b>Research</b>	<b>5</b>
Brief	5
Hypothesis	5
survey results & analysis	6
Market Penetration	8
Social	10
Legal	11
<b>Technology</b>	<b>12</b>
Overview	12
Application	12
Development Tools	12
<b>Design</b>	<b>13</b>
Brief	13
Icon App	13
Icon Explanation	13
iOS System Mockup	16
<b>Branding</b>	<b>18</b>
Brief	18
Our message	18
Our values	18
Our logo	18
<b>Plans</b>	<b>19</b>
Implementation Plan	19
WBS	19
Communication Plan	21
Contingency Plan	22
Test Plan	23
Approach	23
Alpha	23
Demo	23
Beta	23
Training Plan	24
Overview	24

Emphasis	24
Specifics	24
<b>Features</b>	<b>25</b>
Brief	25
Feature list	25
<b>Technical Documentation</b>	<b>26</b>
Brief	26
UML 2.0	26
Class Diagram	27
Use case diagram	28
Entity Relationship Diagram (ERD)	35
<b>Registers</b>	<b>36</b>
References	36
Books	36
Web pages	36
Blog posts	36
Online Articles	36

---

# Introduction

---

## Overview

GigMeln is an application to provide a platform for student casual job seekers and employers to meet.

There is a very high demand for last minute casual jobs that fit with student's schedules and activities, especially amongst the growing international community. Not only this but, the student population in cities is growing at a high rate, and once again, international students are a big factor here again.

The nature of these last-minute casual jobs asks for generally unskilled or semi-skilled population, with 'easy-to-obtain' qualifications from government bodies and/or private agencies.

The world of online ordering and next-to instant delivery is now more present than ever. And in this arena, mobile is king. It is therefore only but natural to couple last-minute casual jobs with the always connected world of smartphones.

We aim for GigMeln to be as intuitive and easy to use as possible while and provide a trusting environment for both users, we definitely want to reflect the casual job vibe through the interaction with the app.

# Research

---

## Brief

Research is imperative to obtaining conclusions which are not based on hunches or guesses.

To confirm our set beliefs, research is the only way to ascertain if such ideas are actually supported to be considered as knowledge.

In order to be efficient in the usage of resources it is essential to do the correct decisions on how and where to apply these very resources.

The very words of Albert Einstein: "*If we knew what it was we were doing, it would not be called research, would it?*"

## Hypothesis

---

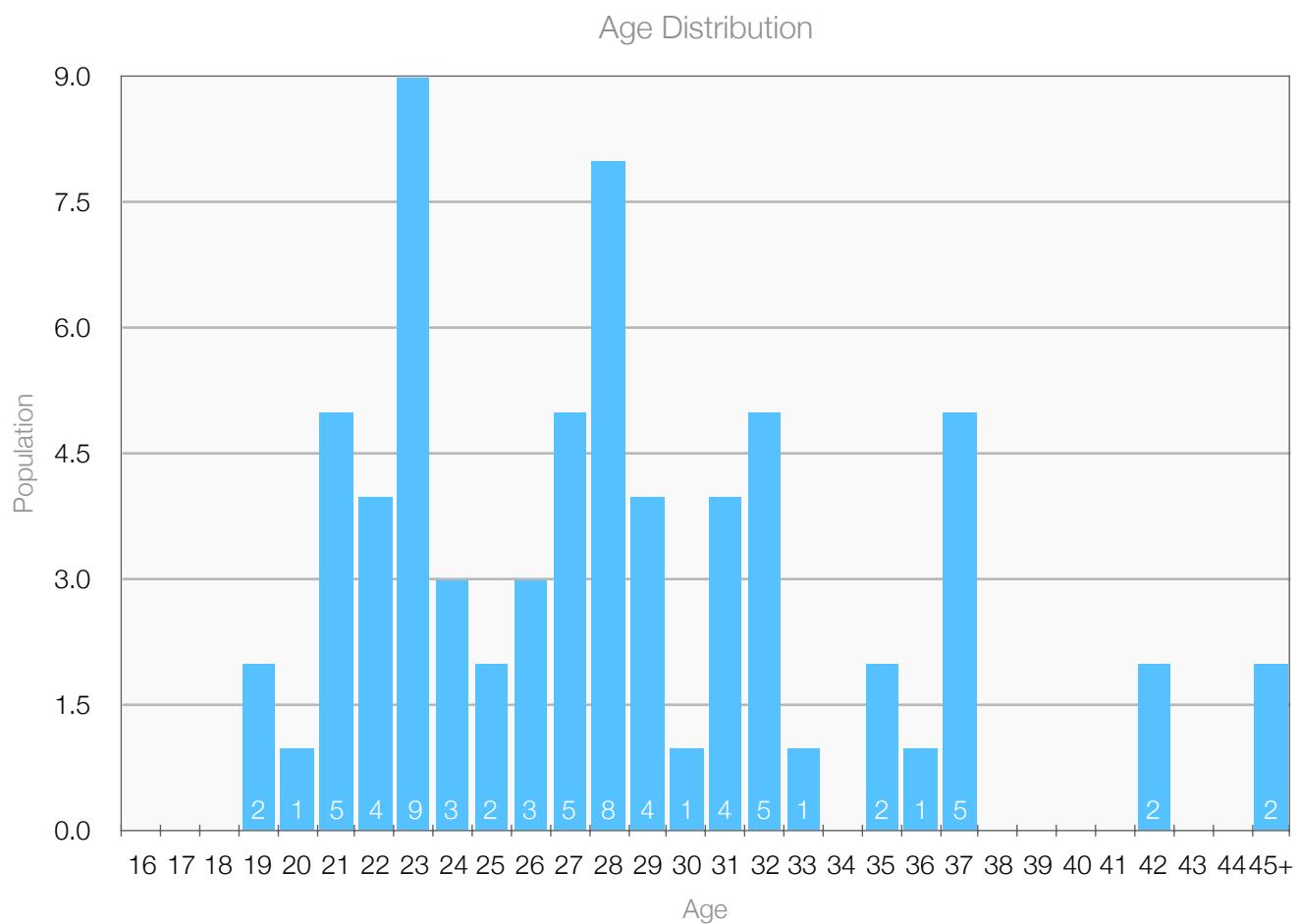
Our belief was that the major seekers and takers of casual jobs are people who:

- Can't commit to a full-time employment due to a multitude of reasons
- Want and/or need an extra income
- Have extra time in their hands yet it is not compatible with standard employment hours
- Have irregular working hours
- Already have a primary job
- Sporadically have moments of low or no-income
- Don't have a permanent status or are travelling

This has led us to understand that the demographic persona will be mostly made of:

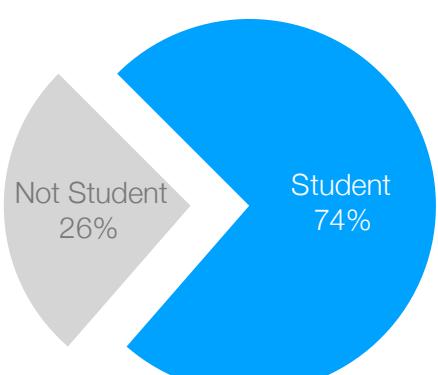
- Students
- Relative young working population when compared to average working population
- Adults without much work experience
- Recently unemployed people looking for another stable job but are caught up in the middle

## survey results &amp; analysis

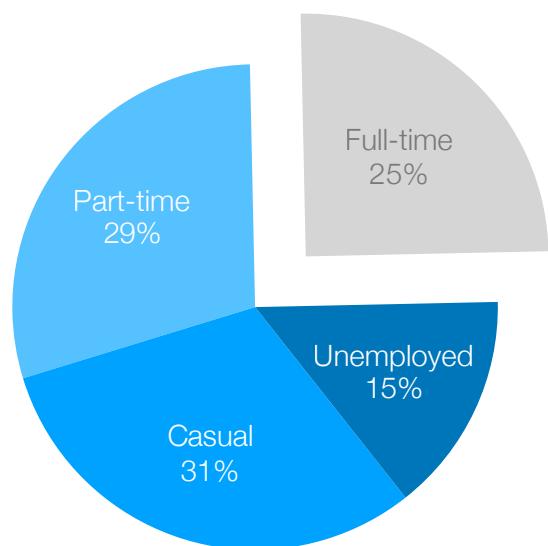


It is easy to verify that the number of respondents is mostly concentrated in the young adult age.

Student Distribution



Employment Distribution



It is also deductible that from that vast majority, is a student, with only 26% of the population not being a student.

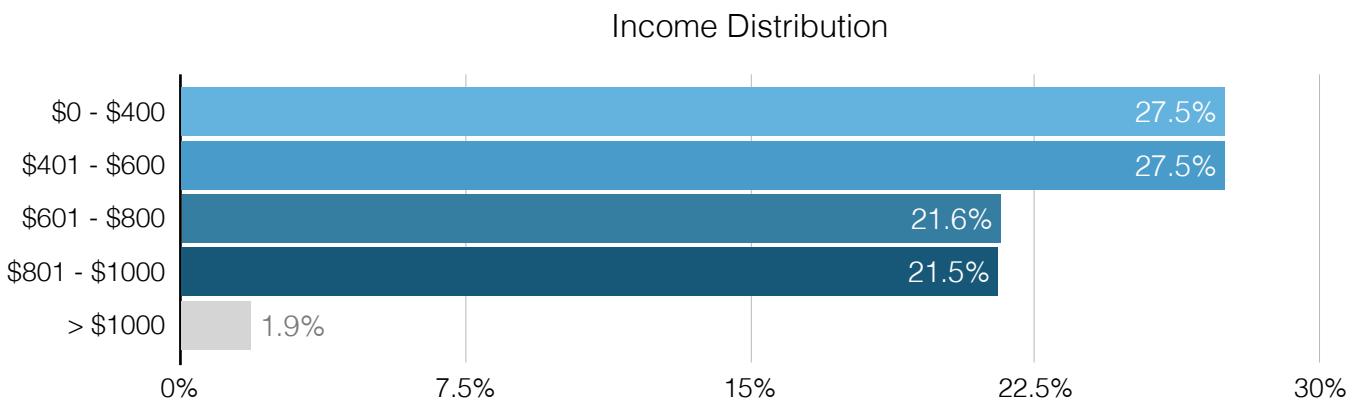
But even more important to note is that from the entire student population, 75% are either unemployed,

casual or part-time, in raw numbers what this means is that from the entire sample of 69 respondents:

- 69 respondents x 74% = 51 are students, and
- 51 students x 75% = 38 are either unemployed, casual or part-time, so
- 38 users who would heavily benefit from our application, this is 55% of our entire sample!
- This goes without saying that casual and unemployed, makeup 46% of the entire student population.

It comes to show how meaningful our application is for these people, and another interesting point is that it is also proof that 31% of students already rely on casual employment, so a big chunk of users who would instantly adhere to our application platform.

Now let's explore the income distribution of these 38 users.



Not surprisingly, the results tell us that the majority of the people are earning less than \$800, and virtually everyone less than \$1000. This clearly displays that there is a really big number of users who would benefit very much from using our application.

The most interesting is that above 50% of the population would be guaranteed major users, since they have a lower income and therefore seemingly more time.

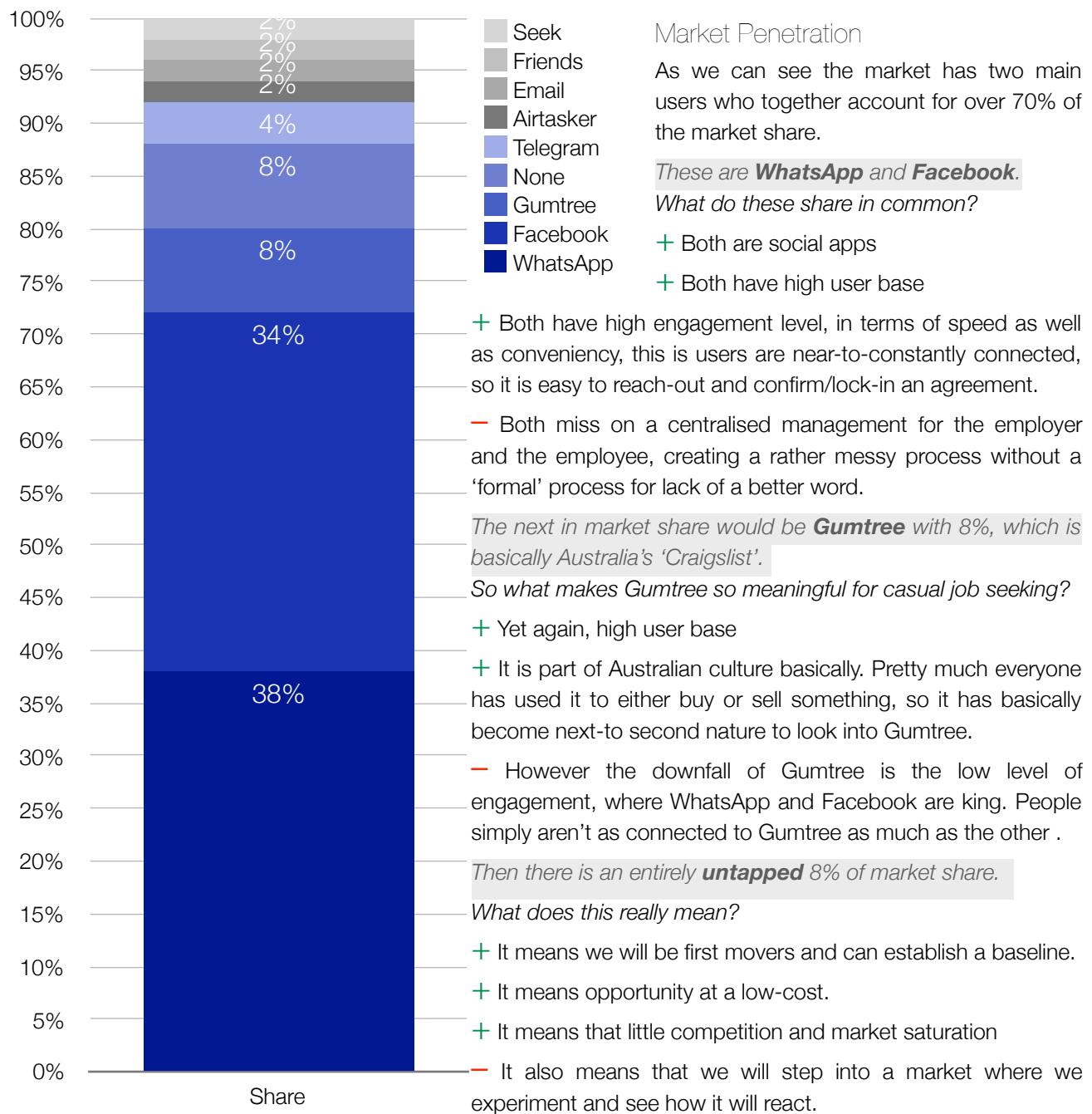
Now we want to understand the current methods being used amongst these users to obtain work.

This is important because of a various reasons:

- Understand which applications are being in use currently
- Understand which features in these very applications make its usage possible for the purpose of seeking work for these users
- Understand what's the market penetration given the current situation
- Understand what's the market's need for a dedicated application
- Understand what's the format that casual jobs are being posted like in the most used applications
- Understand what are similarities between the most used applications
- Understand what are the platforms that we should be targeting for the advertising campaign, during the pre-launch and launch stages, of our application
- Understand what and how do the most used applications lack in features to become a casual job seeking focused platforms
- Understand what niche do the outlier applications fill for their users when seeking casual jobs
- Reach out to both user types of the most used applications to introduce them to our application and showcase how and why we are a great replacement
- Reach out to both user types of the most used applications to build and establish a relationship in order to reach a critical mass number to kickstart/bootstrap

Now if we take the outlier of people who are earning above \$1000 per week, and we look at which applications do they use to obtain their work currently.

### Application Distribution



*Coming up next is Telegram with 4%.*

*So what is Telegram doing differently?*

- + They are doing one thing that was what firstly prompted us to start this project, which we knew that WhatsApp was missing on, unlimited number of users per group, meaning that a larger population has access to more job offers.
- Telegram's main problem is that there is a smaller set of users generally speaking.
- Like every other application, it also doesn't have a centralised management system for applicants, for the very reason that it was never designed for the purpose that it is being used for by this population.

*What about the rest? Airtasker? Email? Friends? and Seek? their representation all put together is smaller than the 'none' section.*

So what does it mean? It means that they are fitting a niche.

- **Email?** It fits more for casual jobs that have a more formal standpoint and are usually with higher pay-rates, for example projects like website creation, more freelancing type of jobs
- **Airtasker?** In this case, what we once had considered to potentially be a competitor candidate, after research we've learnt that it is actually not. Why? Because it really isn't a casual employment, it is more 'paid-help' service, they are generally smaller, of a shorter duration and also behave more in a freelancer scenario. Another big caveat, 'Airtasker' takes a big chunk of the workers income, which at this point in time is mismatching of our business model.
- **Friends?** Friends are a great way, and these tend to be more reliable and of shared experience which is great for everyone involved, representing a higher success rate due to the fact that the person recommending knows the skill set of the referred as well as the needs of the employer. We plan to tap onto these by integrating 'social media' referrals into our application. (please see the dedicated use case page)
- **Seek?** We didn't take that much time to understand 'Seek'. Why? Because we know that its' target audience is entirely different and we would much rather spend our resources in a different manner.

## Social

From a social stand-point GigMeIn has the potential of having such an enormous impact.

Being an application that connects employers and employees together, and more importantly, due to the casual nature of the work that is posted it means that users are going to find themselves using our application very frequently unlike other employment solutions.

For every and any user looking for casual work, it will help to:

- find casual work during less busy periods to earn extra income
- obtain genuine reviews of other workers' experience when working with the company

Specifically, in the students' lives it will help to:

- circumvent the difficulty of finding work that fits with timetables,
- deal with exam periods' need to focus,
- have freedom to focus on assignments and exams during determinate periods, while needing not to worry about job commitments

Regarding the employer, some of the benefits are:

- first and foremost, a centralised ecosystem for all their casual work needs and therefore
- an easy and effective way to reach-out only to a vast population of workers who can actually perform the work set by you and the time set by you
- a platform for obtaining genuine reviews of other employers in a glance to better aid at deciding who to employ

## Legal

To ensure that we have the less amount of legal implications or other barriers when creating the application, we looked into the entire process of obtaining/providing casual work.

After careful consideration, we've decided not to handle some of the most delicate, legal prone aspect, and potentially risky parts but, instead to focus on what we could do really well on the application with little to no repercussions.

Basically we will act as a platform to match and connect both people, the employer and the employee,

We won't be handling payments matters, instead to create trust and protect both the employer and employee, we've decided to implement a work review system instead. To ensure that the system remains meaningful through the user's experience, we are tying each user to an uniquely identifying number either via an ABN for the business, TFN for the employee.

# Technology

---

## Overview

Most applications use a mix of web and mobile apps. There are multiple advantages and disadvantages of using this approach.

Due to the nature of the casual, easy to use, and quick access to the application, we turned to something which people have easy and quick access to and are very comfortable using.

The mobile phone.

We want to reach to the most personal device that people carry around everywhere at any time.

It also happens to be that the mobile phone allows for dedicated applications and direct access by, from and to the users. The ability to use notifications for users to be reached is a major requirement, users need to know as soon as possible any updates of their job posts and job applications.

The mobile phone also carries with itself multiple technologies that may come to be useful later, should we consider other features and benefits to the user, such as GPS, WiFi location, SMS and others.

## Application

---

We want the application to be available on the two major platforms as soon as possible, this would mean iOS and Android.

The application should take advantage of the operating system's native elements and components, this is to guarantee that the users of each specific platform are comfortable and expectant of the position, look, feel and behaviourism of said elements and components.

## Development Tools

---

We've looked into multiple development environments, frameworks and other technologies to aid us in developing the application in the most seamless way, given the cross platform necessity. After pondering over Xamarin, Electron, React, Ionic and other few strong candidates, we ended up choosing the go with Ionic. While the deployment would be done with Apache Cordova.

Another point that was important for us was the cost of development, Ionic is free with a freemium plan, which we won't be needing up until a later stage possibly.

At a later moment in time we have been looking into using Firebase as the online storage of our application.

We will be using either GitHub or BitBucket as an online repository for our project.

In regard to design files and documents, outside of using the native elements, mockups and sketches will be done in an application like Adobe XD, Sketch, or Photoshop we aren't quite sure just yet, we need to do further considerations.

For documentation we've decided to use iWorks suite either as an application on iCloud or standalone. This is because of the ease of use and collaboration feature, and it was preferred to using Google Drive or Dropbox.

We have considered using Cloud9 for collaborative development, or use Atom's collaborative modules for collaborative development.

We will be using GitHub Pages or another similar service to be able to demo our application and use it to gather feedback from user testing.

All of this is obviously open to change and modifications as we may very well discover better catered services that fit better our development needs.

# Design

---

## Brief

We cannot underestimate the importance of design in this day and age.

The purpose behind design is to translate an idea to a final product that reflects its value in the user experience.

We've began to draw the use case scenarios, the benefits and features of the application and now its time for the ideation to start taking shape through the usage of lines, colours, contrast and other elements to produce the application's layout and interface.

The layout and interface is to be as familiar and simple as possible to our target audience, this will cut on learning curve and make for an intuitive experience, in order to generate the intended use, all the while, delivering the solution to the predefined problem we had set to solve.

As a discipline, design is to plan the creation of a solution with the intention of improving human experience with respect to a specified problem, keeping in mind at all times the concept of user-centeredness, that is at the heart of the design thinking movement.

Design is about progress. It is important to never lose sight that it is an ever-going process, where at each particular instance, the design effort takes new considerations.

## Icon App

---

An icon for an application is extremely important as it is the first moment of engagement with the audience. After a long research and obtaining insight from a knowledgeable individual who forwarded us to a set number of extremely eye-opening articles, one that has proved to be very valuable in the deciding of the logo was (<https://www.canva.com/learn/what-is-the-golden-ratio/>), upon careful reading we've drawn an overlay that helped with the design of the icon logo.

It helped creating symmetry and maintaining a more concise designed that kept its proportions following the golden ratio rules as seen here (<https://www.vecteezy.com/vector-art/91542-golden-ratio-proportions>).

Also we took into account the guidelines set by the platform that we've targeted for our application's design, we want to make sure that the design of the application marries as perfectly as we possibly can with the native experience.

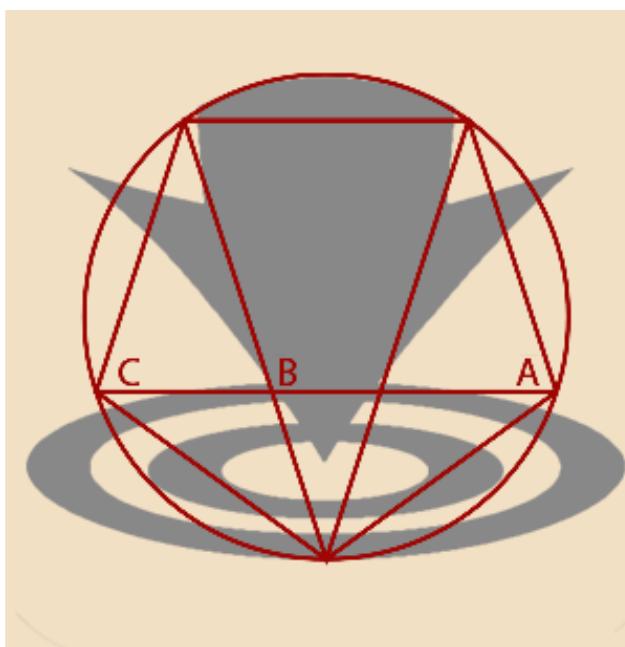
## Icon Explanation

Our goal was that the icon would tell as much as possible about the application and its use. After long hours of thought and discussion we came to agree that we wanted to invoke in the icon the idea of:

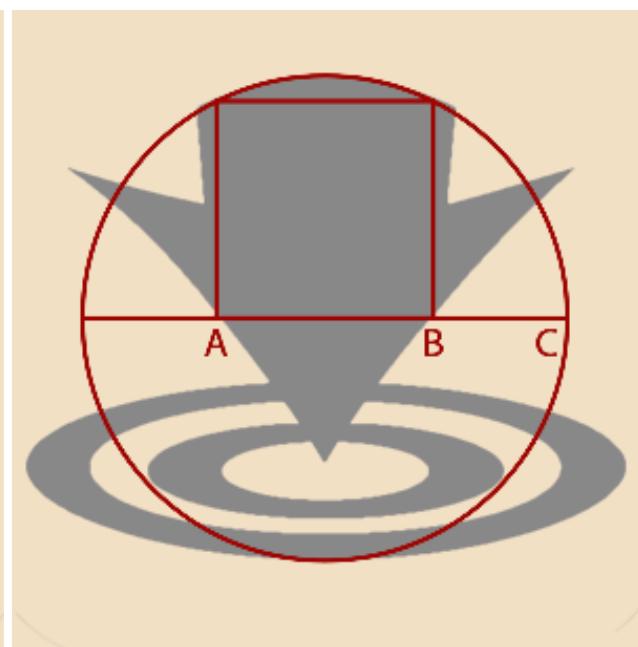
- Speed. As in, how quickly you'll be encountering casual work.
- Location. As in, every job has a place where it takes place.
- Now. As in, you choose at this very moment to search, find and apply for work.

It also had to contain a great part of the middle of the icon area, yet we didn't want it to become cluttered. So it seemed to us that a good use of the real estate and the white space of the icon.

## Golden Ratio Icon Design

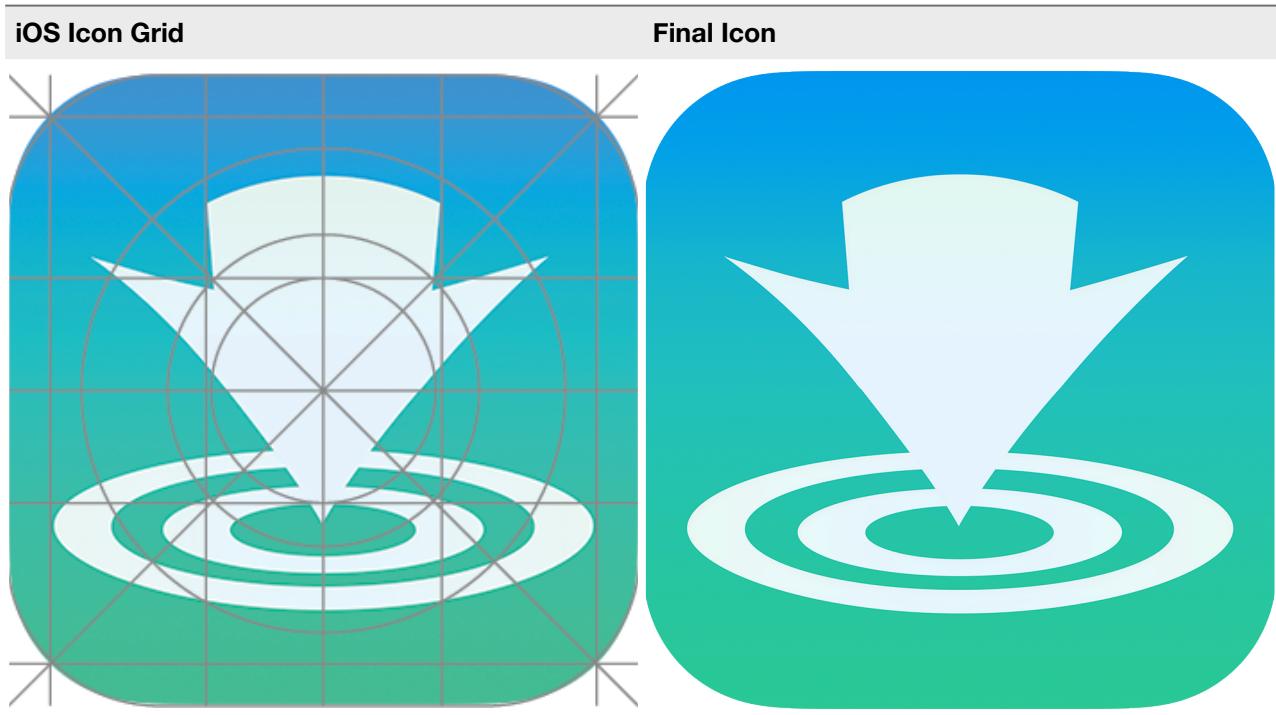
**Home Screen**

This overlay has helped creating the distance between the very top and bottom of the icon and maintaining the two elements separated on its suggested proportions. It has also helped creating designing the arrow angles.



This overlay has helped creating the distance between the arrow top lines and establishing a filled area within the block of A-B.

## iOS Icon Grid



This overlay establishes clear guidelines.

We can centralise the icon with the correct spacing between the top and bottom.

Maintain the icon outside of the margins for the icon to ensure a more standardised and cohesive design with the ecosystem's look and feel.

Another great point is the use of the whitespace between the shapes while maintaining it still full to

After careful considerations and analysing current trends, both from third parties and native operative system's applications we noted the value of gradients and subtle near seamless changes.

The teal and blue colours were chosen for their close relationship and similar transitional colour.

The low-transparency white was chosen because a full flat white produced too strong of a contrast, and that was disturbing to the eyes.

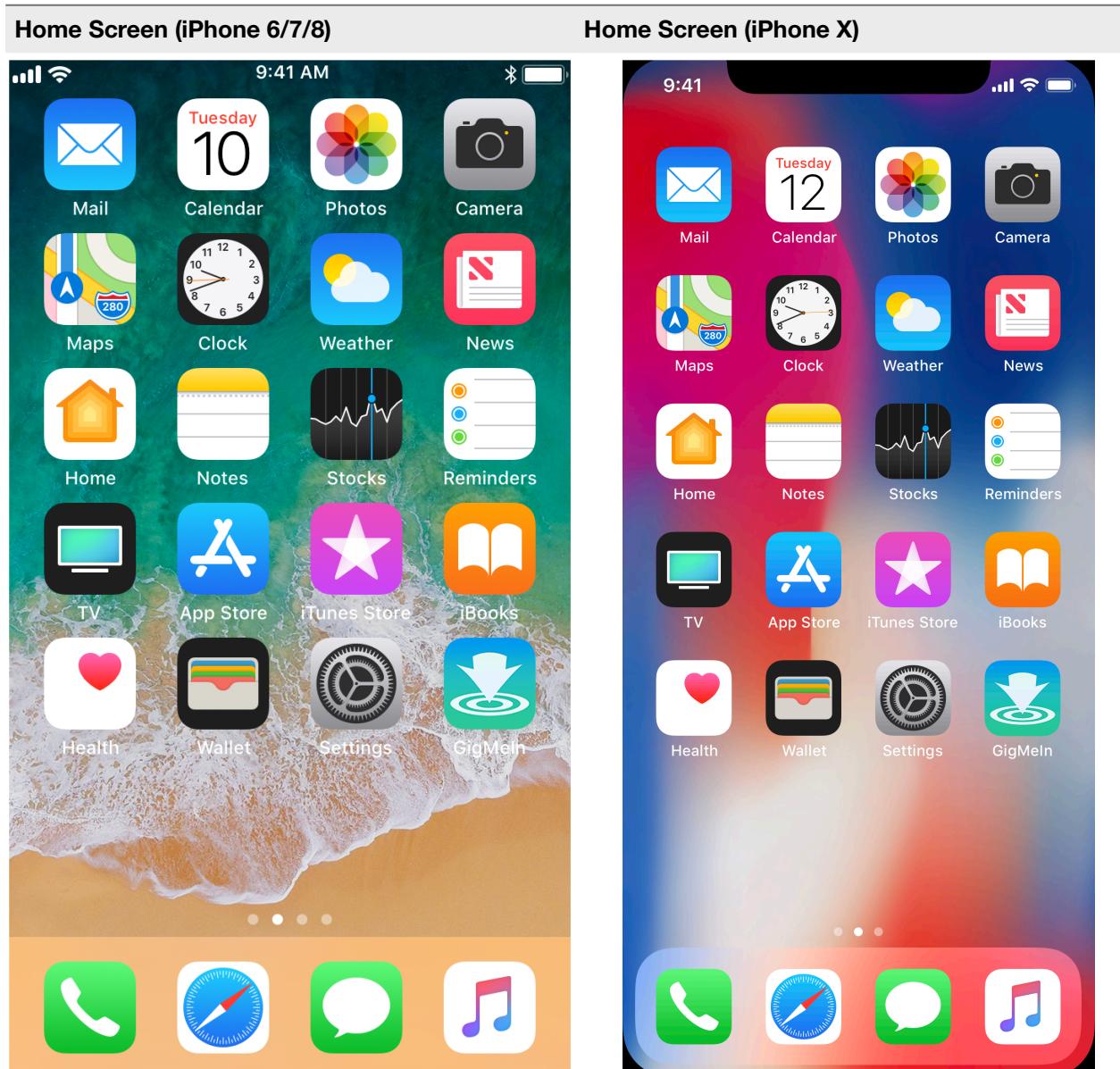
## iOS System Mockup

It is also important see exactly how will the icon look in the every day use for the user, therefore quite a lot of testing went into getting the right feel by working on the proportions, the colours, the transition and the animation of the opening and closing of the application.

We've tested on both iPhone standards. We are missing on the iPhone (6/7/8 Plus) versions however it is mostly because the change is not very pronounced on the 'Home Screen'. We are quite happy with the outcome.

Later on we might come to change it but for the time being this version seems to be the best possible given the current time constraints.

### Mockup Explanation



The logo upon finished design we have put it on the 'Home Screen' to see how well would it fare with the other applications next to it and if it would be easy on the eye or not.

We applied on the iPhone X to see how well it would also fare against different background colours.

## Notifications

Simple Notifications Screen	Notifications Expanded Screen

We've also created a set of sample notifications that would resemble what it would be like on the user's experience on their device on a day-to-day basis.

# Branding

---

## Brief

At no other moment in history has branding been as important as now. With how easy today is to reach people, it is crucial to reach them in the most clear and meaningful manner.

After all, yelling isn't the most effective form of communication, and neither is to just put your brand out there. The message has to be thought out and to be delivered, and just as important, different messages belong in different channels, or as we call them today, platforms.

### Our message

As previously stated, GigMeln purpose is to provide a community driven platform to connect casual employers to semi-skilled workers and students that can't commit to a regular schedule.

### Our values

<b>Value</b>	<b>Short description</b>
Community	A strong community is the key to keeping a well maintained platform. It drives to its members meaningful information. Ultimately it is users who dictate the lifetime of any service or product as they are the ones who keep it alive. The proof of this is Reddit.
Trust	A great part of the community value is that it is the very engine that drives trust.
Design	We respect users time and as users ourselves we also understand the frustrations that come from using poorly designed applications. Our intent is to create and implement the very best.
Delivery	We want to show to users that we intend to deliver on our promise
One of them	We want to show that we are also one of them. That we understand their frustrations and needs because we've developed this product while being in their shoes.

### Our logo

The thought behind the design of the logo was based on what we plan to deliver with GigMeln's platform.

The logo is meant to show to users what GigMeln is at its core:

- An easy-going attitude and usability
- All work is both casual and location based
- A modern take on connecting casual workers and employers
- A direct and simple way to apply for work
- An honest approach to employer/ee review feedback
- A personalised ever-learning tool that helps you find more meaningful and better employment based on profiling
- A tool that advises based on thousands of users and jobs on what best set of skills, licenses and/or courses could benefit you
- No middle-man taking a cut of your pay, it all just between you and the employer
- Central and easy way to manage casual work and workers



# Plans

---

## Implementation Plan

### WBS

This project is split in 3 phases. phase 1 will happen during the Advanced Studio 1 semester whereas the other 2 will happen during the Advanced Studio 2 semester.

Project Implementation Timeframe

Semester 1				Semester 2			
Nov	Dec	Jan	Feb	Mar	Apr	May	Jun
Phase 1		Phase 2		Phase 3			

Phase 1 WBS

WBS	Description	Duration	Delivery Date
	Start of Phase 1	0 weeks	2017-11-03
0.1	Idealisation	2 weeks	2017-11-17
0.2	Project Mindmap	1 week	2017-11-24
1.1	Use cases	2 weeks	2017-12-08
1.1.1	Use case diagram	1 week	2017-12-15
1.2	Class diagram	1 week	2017-12-22
1.2.2	Use case specification for complex features	1 week	2017-12-29
1.3	Documentation	3 weeks	2018-01-19
1.4	Mockups design	2 weeks	2018-02-02
1.5	Project presentation	2 weeks	2018-02-16

Phase 2 WBS

WBS	Description	Duration	Delivery Date
	Start of Phase 2		2018-03-12
2.1	Development environment setup	1 week	2018-03-19
2.2	Database setup	1 week	2018-03-26
2.3	User interface implementation	4 weeks	2018-04-23
2.4	Features implementation	5 weeks	2018-05-28
2.5	Testing	1 week	2018-06-04

Phase 3 WBS

WBS	Description	Duration	Delivery Date
	Start of Phase 3		2018-06-04

<b>WBS</b>	<b>Description</b>	<b>Duration</b>	<b>Delivery Date</b>
3.1	Testing with potential users	1 week	2018-06-11
3.2	Adjustments based on feedback	2 weeks	2018-06-25
3.3	Final presentation	1 week	2018-07-02

## Communication Plan

A successful project requires a well-planned communication between stakeholders of the project. More even when we take into account the development methodology used on this application.

In the following matrix it will be detailed the level of communication expected

### Blog

<b>Description</b>	Publicly communicate the progress of the project			<b>Responsible</b>	Hugo Santos
Frequency	Even Weekly	Channel	Web	Channel	Web
Target	Public	Access	External		

### Status Report

<b>Description</b>	Status update and research findings			<b>Responsible</b>	Hugo Santos
Frequency	Uneven Weekly	Channel	Document	Type	Formal
Target	Stakeholders	Access	External		

### Cloud Storage

<b>Description</b>	Share of project files amongst the team			<b>Responsible</b>	Hugo Santos
Frequency	Weekly	Channel	GitHub Google Drive	Type	Formal
Target	Team	Access	Internal		

### Trello

<b>Description</b>	Project management software Features and development issue tracking			<b>Responsible</b>	Hugo Santos
Frequency	Weekly	Channel	Web	Type	Formal
Target	Team	Access	Internal		

### WhatsApp

<b>Description</b>	Instant messaging application			<b>Responsible</b>	Hugo Santos
Frequency	Daily	Channel	Web	Type	Informal
Target	Team	Access	Internal		

## Contingency Plan

Writing a contingency doesn't sound like a very hopeful message, yet it still is important to do, and taking into account the harsh reality of most projects nowadays, I may even say it is a necessary step.

Already on our development we've implemented a tiered, levelled and prioritised set list of feature-benefits which we can fall to on the occasion that we may need to trim the least necessary / important features from the project if it comes to that.

Risk analysis matrix

<b>Likelihood</b>	<b>Consequences</b>		
	Minor	Moderate	Major
	<i>demand small changes on the project (up to 2 features)</i>	<i>demand big changes on the project (more than 2 features)</i>	<i>project survival is at risk or demand changes in the whole project</i>
Likely (>= 30% chance)	2	3	4
Moderate (30% >= 5% chance)	1	2	3
Unlikely (< 5% chance)	1	1	2

Risk analysis

<b>Scenario</b>	<b>Likelihood</b>	<b>Consequences</b>	<b>Risk Level</b>	<b>Action Plan</b>
Scope is too big	Unlikely	Major	2	Make and execute a good planing of the project.
User acceptance very low	Moderate	Major	3	Make research to design something that meet what the users would like to use.
Legal issues	Unlikely	Moderate	1	Research about legislation in regards of jobs apps.
Gold plating	Moderate	Minor	1	Stick to the project Scope; Design emphasis placed on minimalism.
Lack technical skills	Unlikely	Minor	1	Use well known and consistent technologies to develop the project.
Lack of communication	Unlikely	Minor	1	Use channels that can be used and accessed easily.
Unachievable schedule	Likely	Moderate	3	Schedule is reestimated several times over the course of the project.

# Test Plan

Testing is of the utmost importance for creating a robust and performant application.

## Approach

---

### Alpha

This is an in-house test. We will be running and using the application to test towards the use-cases that we have established, however it is a biased test and we are unlikely to use the applications in ways that it wasn't designed, and there is no chance we could possibly cover every possible scenario. Yet, it is still necessary to get this testing phase done.

### Demo

In-between Alpha and Beta stages, a demo environment will be created for the purpose of showcasing to stakeholders the state of the application, this is important to happen prior to have it become available to public testers and critics who would drive an opinion that might not align with stakeholders' needs, therefore this needs to be firstly approved.

### Beta

This is a public test. We will invite a certain number of users to test the application without any of our intervention so we get as unbiased as possible feedback, this is done because we won't be near the real end users when the application is launched.

The application has to be robust and solid on its own without the need of support, outside of the expected minor bugs.

# Training Plan

## Overview

---

Educating users to use the application is paramount to obtain the desired behaviour from the users. This will also ensure that they take the most benefit out of application through a more efficient use.

Rather than having the user go through a first session explaining the full scope of the application we rather take a game design approach, I will further explain it in the coming paragraphs.

## Emphasis

---

The game design approach is to show the user only what he needs when he needs in a modularised manner. In a more descriptive way, a player only is introduced to a certain ‘mechanic’ (*a feature in the case of applications*), upon the moment that he will require to know it, in order to progress through the ‘level’ (*the use case when speaking in application terms*).

Generically speaking, short tutorials that don’t flood the user with too much information, as too much information leads users to discontinue using the application as the user’s input and the application’s output value to the user becomes imbalanced for the user to see its benefit.

Moreover, game design explores feedback loops better than any other form of design, this has been verified and tested multiple times over, both, imperially and statistically, the retention and engagement levels of video games has been proven. So much so, that we’ve been seeing the growth of gamification being explored in various scenarios, such as raising awareness to topics of interest by governmental institutes, advertisement and marketing companies. Pretty much every industry has taken notice of game-design faculties and is implementing it in its products, services, processes, even training.

We plan to make implement in the same fashion.

## Specifics

---

*Plans will be updated when a more concrete plan with specifics has taken place.*

# Features

---

## Brief

The table below is a list of the features that will be implemented for our software. It is divided in 3 columns. The first is the *priority*, which means how important is that feature for our software relative to the others. The second column is the *description* of the feature which provide a title that explains a little of what that feature is. The tier *column* shows how is the implementation breakdown. Almost as a WBS in a table. Basically shows the dependency of the features. Higher the number more dependencies it has. If the above feature tier number is smaller then the current feature tier number, it means that the current feature depends on the feature above and so on. If they are equal they are in the same level. And for last if they are higher it means it depends on the above feature that has the first number smaller than the current feature.

## Feature list

Priority	Description	Tier
	<b>Authentication</b>	<b>A</b>
	Login as an employee	1
	Login as an employer	1
	<b>Employer</b>	<b>B</b>
	Job post	1
	View posted jobs	2
	View job applicants	3
	View applicant details	4
	Accept job application	5
	Review employee	6
	Edit profile	1
	<b>Employee</b>	<b>C</b>
	View listed jobs	1
	Apply for a job	2
	Reject a job	2
	Super application	2
	View job details	2
	View job applications	2
	View employer details	3
	Edit profile	1

# Technical Documentation

---

## Brief

This section shows all the relevant documentation for the application proposed in this document. We used UML 2.0 (Unified Modelling Language) for most of the documents which is standard in the software industry for project documentation. For the database design we used ERD (Entity Relationship Diagram) which is also a standard for database design.

### UML 2.0

---

UML was created by Jim Rumbaugh, Ivar Jacobson and Grady Booch at Rational Software and it was adopted as a standard in 1997 by OMG (Object Management Group) with the purpose to define a common design language to model software applications. In 2005 ISO (International Organisation Standardisation) also recognised UML as a standard empowering IT professionals to read and spread their design plans (Wikipedia 2018).

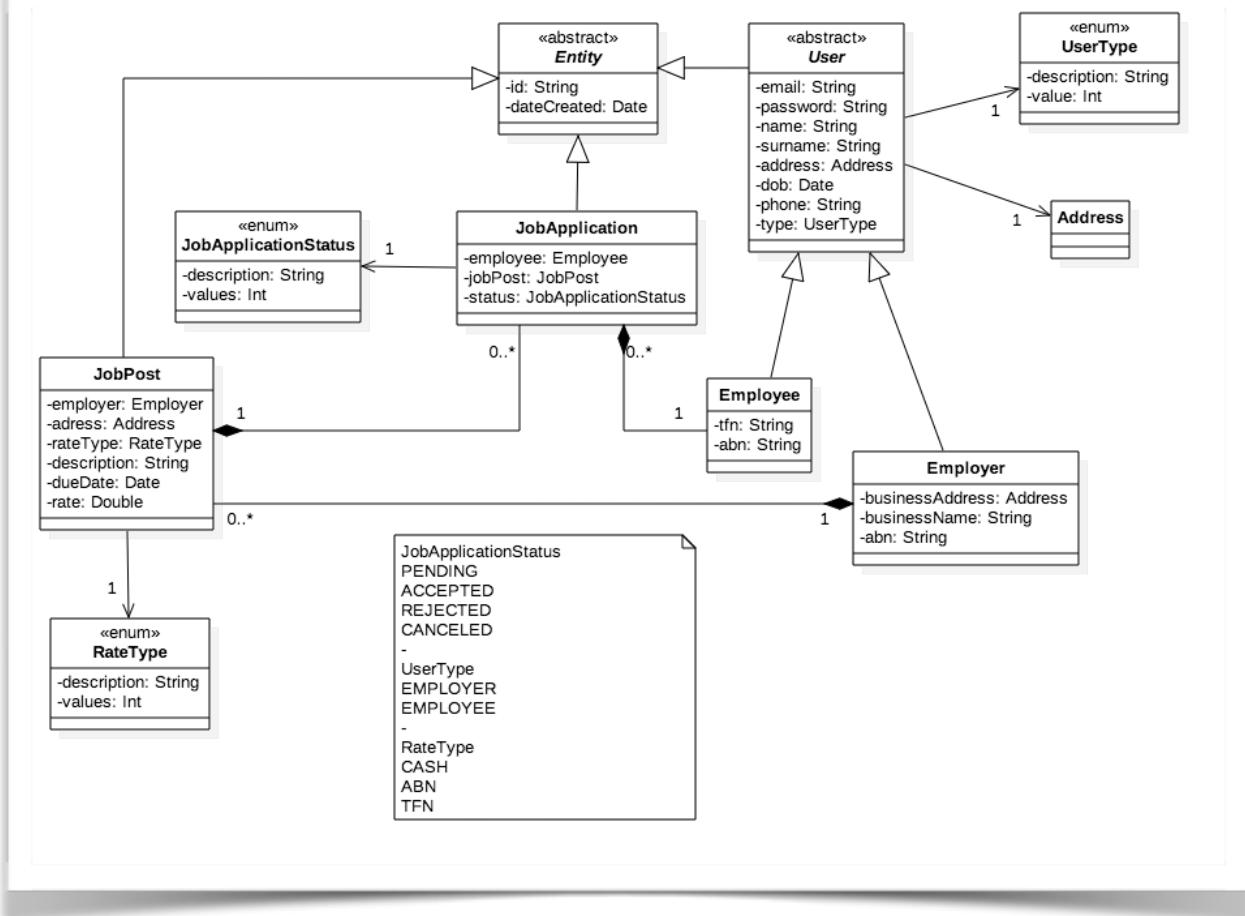
The main reason why UML has become a standard is because it is language independent. It means that it can be used to design software disregarding the language it's being used (Donald Bell 2015). e.g.: Java, C#, Ruby, Python and many other languages.

There are several UML diagrams that could be used to design this project but not all of them are needed to successfully document a software design, is up to the designer to choose which of the diagrams are more suitable for his/her project.

For this project has been chosen only two UML diagrams for an initial documentation, which are the Class Diagram and Use Case Diagram, following by the Use Case Specification. There is a brief description of each one of these in the next chapters.

## Class Diagram

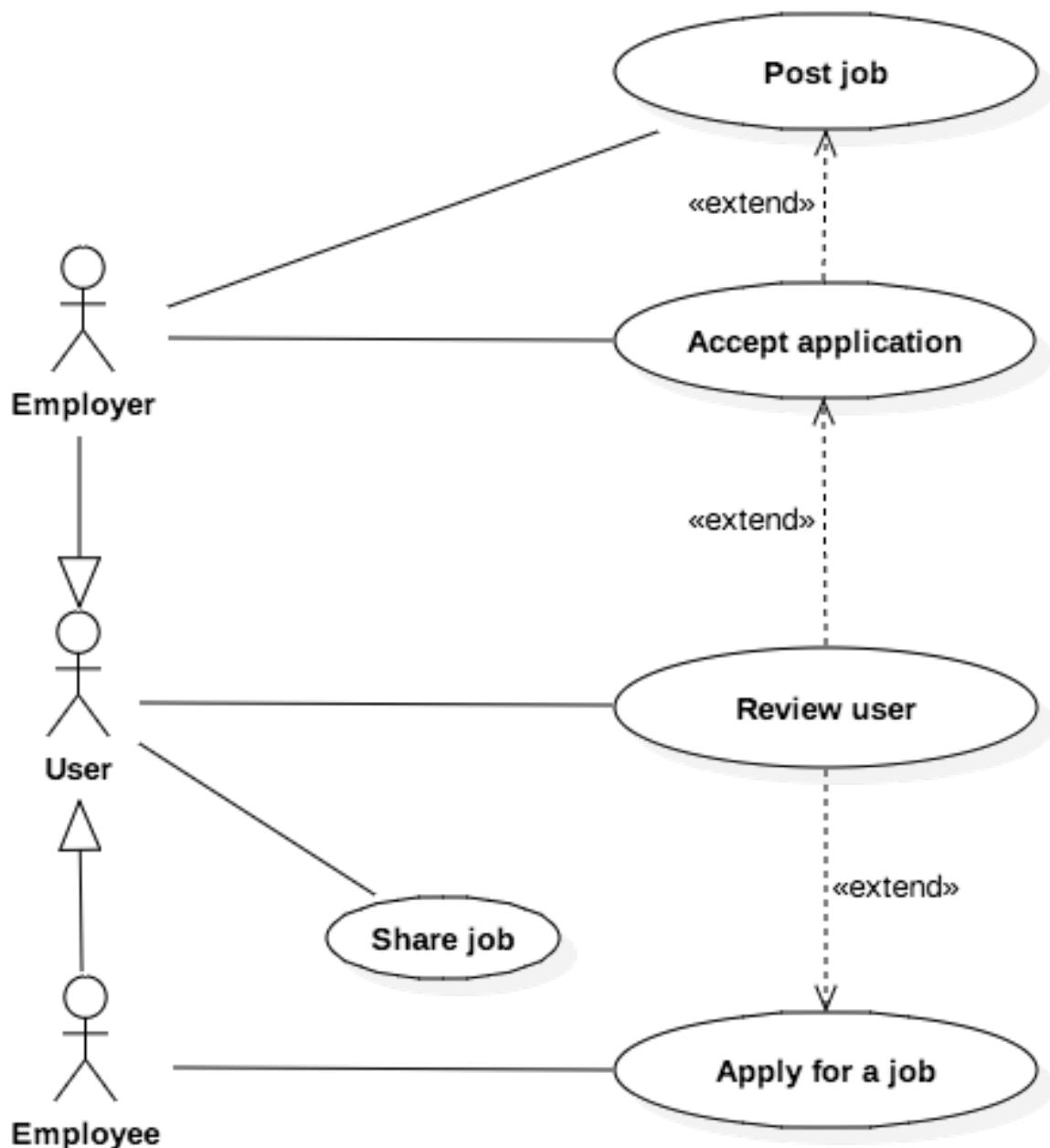
The purpose of the class diagram is to show how the classes of the application are structured. According to Pilone and Pitman this is one the most fundamental diagrams in UML. It is used to capture the static structure of a software or, in other words, how things are put together (Pilone & Pitman 2005, p.11). The classes within the class diagram represent not only the classes to be programmed but the interactions between them.



## Use case diagram

The purpose of a use case diagram is to represent the interactions of the user and the system by identifying and defining the most elementary business process of a system. Each use case can be drilled in much more detail when the use case specification is done in together with the diagram itself (Statzinger & Jackson, 2008).

We decided to include in our diagram only the most relevant use cases. Including all minor features as a use case will only consume time and will not aggregate much value to the application.



## UC-01 - Post Job

**Primary Actor:** Employer.

**Goal:** Post a job advertisement.

**Scope:** A search for job application.

**Brief:** As an employer I post a job so other users (Employees) can see it in their list of available jobs.

**Success guarantees:** The job is shown in the Job Posts list of the employer and is available for application by the employees.

**Preconditions:** The user must be signed up as an employer.

**Triggers:** The employer invokes a new job post request.

**Basic flow:**

1. The system display a card with placeholder text in the required fields for a job post. Within the card will be displayed a Text Box for the title, a Text Area for the job description, a pick calendar for the dates, a field for the start and finish time, and a field for the hourly rate.
2. The employer selects on the title placeholder.
3. The system clear the placeholder and enable the field for editing.
4. The employer inputs the title and selects OK.
5. The system display the card with the updated information.
6. The employer selects the description.
7. The system clear the placeholder and enables the description text area.
8. The employer input some text in the description and selects OK.
9. The system display the card with the updated information.
10. The user selects the image.
11. The system request the image location.
12. The employer selects the image from his phone or web and selects OK.
13. The system display the card with the updated information.
14. The employer selects the date field.
15. The system display the calendar of the current month.
16. The employer tap on the dates he needs and selects OK.
17. The system display the card with the updated information
18. The user select the hourly rate.
19. The system display a spinner for the hourly rate.
20. The user selects the hourly rate and selects OK.
21. The user selects finish.
22. The system validate if all the required fields are filled.
23. The system offer the premium advertisement.
24. The employer accepts the premium advertisement and click finish.
25. The system saves the job post and the notify possible interested employees about the post.
26. The system display the view with the list of posted jobs by the employer.

**Extensions:**

2-24.a. *Cancel the post*

1. The user select cancel.
2. The system discard all the inputed information, then go to step 12.

22.a. *Input validation*

3. The user don't input any information in the corresponding *n* step and click next.
4. The system display a validation message, then keep in the same step.

23.a. *Not premium*

1. The user dos not select premium.
  2. The system saves the job, then go to step 26.
-

---

UC-02 - Apply for a job

**Primary Actor:** Employer.

**Goal:** Submit a job application.

**Scope:** A search for job application.

**Brief:** As an employee I submit a job application so employees can accept me or not.

**Success guarantees:** The employee who posted the job gets notified and the employee is shown in the list of applicants for that job and the job is shown if the employee's list of job applications.

**Preconditions:** The user must be signed up as an employee.

**Triggers:** The employee opens the app.

**Basic flow:**

1. The system displays a card with the relevant job information with include: Title, first 20 words of description, hourly rate, date and image.
2. The employee swipe the card to the right.
3. The system adds the job to the applied jobs list, then go to step 1.

**Extensions:**

2.
  - a. Refuse job
    1. Employee swipes to the left.
    2. System discard job, then go to step 1.
  - b. Quick application
    1. The employee swipes up.
    2. The system shows a text area for a custom message for the employer.
    3. The employee selects send

## UC-03 - Accept application

**Primary Actor:** Employer.

**Goal:** Accept an employee for a job.

**Scope:** A search for job application.

**Brief:** As an employer I want to be able to select an employer from a list of applicants to do a job.

**Success guarantees:** The employee gets notified of the employers choice and the job post has its status changed to ACCEPTED.

**Preconditions:** The user must be signed up as an employer. The job post must have received at least one application.

**Triggers:** The employer selects the job post

### Basic flow:

1. The system display all the current applications for the selected job post.
2. The employer selects accept in a specific employee card.
3. The system notifies the employee about the acceptance and makes the employer contact details available for the employee and vice-versa.

### Extensions:

None.

## UC-04 - Review user

**Primary Actor:** User.**Goal:** Review the user who posted or performed the job.**Scope:** A search for job application.**Brief:** As a user I want to be able to post reviews about the person who I'm doing business with after the job is one.**Success guarantees:** The user has been offered the option to post a review for a job.**Preconditions:** The user is signed up and has completed a job or had a job post completed by someone.**Triggers:** The user opens the app.**Basic flow:**

1. The system identifies if the due date of the job and opens a pop-up asking the review from the user.
2. The user writes a review and select Done.
3. The system saves the review and closes the pop-up.

**Extensions:**

2.a. Cancel the review

1. The user select cancel.
2. The system closes the pop-up.

2.b. Submit without all the inputs

1. The user leaves the review empty and selects Done.
2. The system validates the field and display a message warning about the empty input, then go to step 2.

## UC-05 - Share job

**Primary Actor:** User.**Goal:** Share the job post through other platforms.**Scope:** A search for job application.**Brief:** As a user I want to be able to share the job application with friends to multiple platforms to advertiser or recommend to friends.**Success guarantees:** The job has been shared and displayed correctly in the platform the user is trying to share the job post.**Preconditions:** The user is signed up.**Triggers:** The user opens the job details.**Basic flow:**

1. The system display the job details.
2. The user selects the share option.
3. The system display a pop-up showing the supported platforms to share.
4. The user select which platform he wants to share.
5. The system redirects him to the contacts list of the app the user has selected.
6. The user selects in which conversations to share and selects Done.
7. The system closes the pop-up and display the job details page.

**Extensions:**

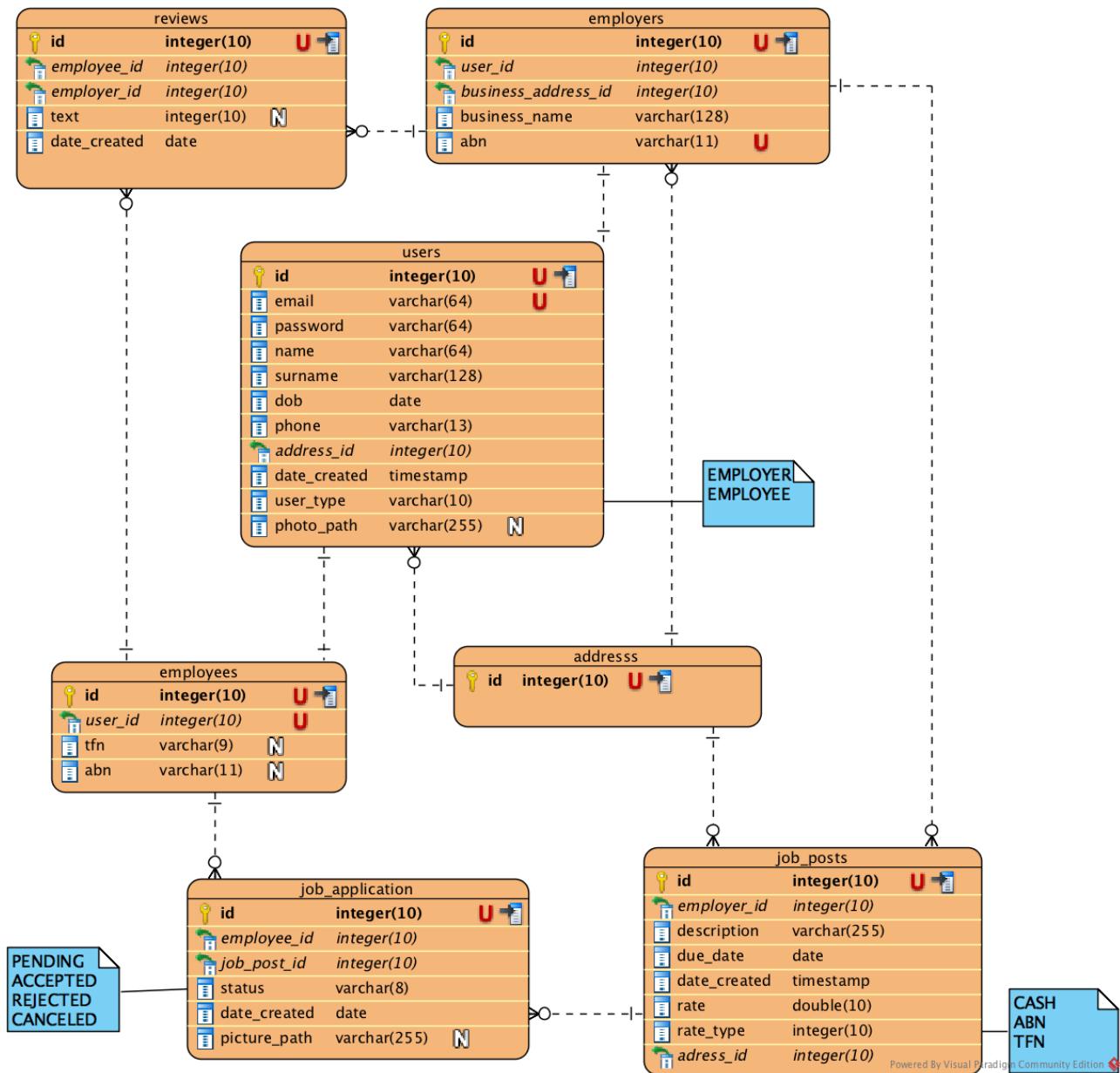
2-5.a. Cancel

1. The user select cancel.
2. Go to step 7.

## Entity Relationship Diagram (ERD)

Originally developed by Peter Chen the idea of Entity Relationship diagram was published in 1976. It is mostly used to define and document the thing that are important to process in an area of business presenting the business data structure in a graphical form.

Below there is the initial ERD model we developed for this project with enough detail to understand how the application will work. Usually Entity Relationship models tend to grow as the business or software gets more complex. It is important to understand that this diagram may have changes during the development process since not all scenarios can be predicted during the design phase.



# Registers

---

## References

---

### Books

- Pilone, D & Pitman, N 2005, UML 2.0 In a Nutshell: A desktop quick reference, O'Reilly Media, Inc; 2nd edition.
- Statzinger, JW, Jackson, RB & Burd SD 2008, Systems Analysis and Design in a Changing World, Cengage Learning EMEA.

### Web pages

- Gal Rimon, '10 Surprising Benefits Of Gamification', 20 July, 2016, viewed 22 January 2018 <<https://elearningindustry.com/10-surprising-benefits-of-gamification>>.
- Christopher Pappas, 'The Science And The Benefits of Gamification In eLearning', 2 December, 2015, viewed 23 January 2018 <<https://elearningindustry.com/science-benefits-gamification-elearning>>.
- Bell, D 2015, 'An introduction to the Unified Modeling Language' IBM, 15 June, 2003, viewed 7 February 2018 <<https://www.ibm.com/developerworks/rational/library/769.html>>.
- Wikipedia 2018, viewed 7 February 2018, <[https://en.wikipedia.org/wiki/Unified\\_Modeling\\_Language](https://en.wikipedia.org/wiki/Unified_Modeling_Language)>.
- iOS Design Themes, viewed 20 January 2018, <<https://developer.apple.com/ios/human-interface-guidelines/overview/themes/>>.
- Interface Essentials, viewed 20 January 2018, <<https://developer.apple.com/ios/human-interface-guidelines/overview/themes/>>.
- User Interaction, viewed 20 January 2018, <<https://developer.apple.com/ios/human-interface-guidelines/user-interaction/>>.
- Image Size and Resolution, viewed 20 January 2018, <<https://developer.apple.com/ios/human-interface-guidelines/icons-and-images/image-size-and-resolution/>>.
- Visual Design, viewed 20 January 2018, <<https://developer.apple.com/ios/human-interface-guidelines/visual-design/>>.
- App Store Marketing Guidelines, viewed 20 January 2018, <<https://developer.apple.com/app-store/marketing/guidelines/>>.

### Blog posts

- Vimby, 2017, 'The Importance of Design'. web log post, June 28, 2017, viewed 8 February 2018, <<https://www.vimbydesign.com/theimportanceofdesign.html>>.

### Online Articles

- Monica Wells, 'Top 10 best examples of gamification in business' School of Game Design, 20 August 2015, viewed 25 January 2018, <<https://www.mycustomer.com/community/blogs/monicawells/top-10-best-examples-of-gamification-in-business>>.
- 'How Shigeru Miyamoto designs a video game' School of Game Design, 'not specified', viewed 25 January 2018, <<https://schoolofgamedesign.com/project/shigeru-miyamoto-video-game-design/>>.
- 'Six Reasons Why Research Is Important' Trinity Institute Dwarka, 20 January 2017, viewed 2 Feb 2018, <<https://www.linkedin.com/pulse/six-reasons-why-research-important-trinity-institute-dwarka/>>.
- Gliddon, J 2016, '7 Reasons Why Research Is Important' Leann Zarah, 27 December 2017, viewed 5 Feb 2018, <<https://owlcation.com/academia/Why-Research-is-Important-Within-and-Beyond-the-Academe>>.