

# Niceway.to Design Specification

Craig Knott  
Version 1.0  
October 7, 2015

## Contents

<b>1</b>	<b>Requirements</b>	<b>2</b>
1.1	Functional Requirements . . . . .	2
1.2	Non-Functional Requirements . . . . .	2
<b>2</b>	<b>Annotated Designs</b>	<b>3</b>
2.1	Design One . . . . .	3
2.2	Design Two . . . . .	3
2.3	Design Three . . . . .	3
2.4	User Flow Diagram . . . . .	3
<b>3</b>	<b>Market Research</b>	<b>3</b>
3.1	Analysis Of Similar Applications . . . . .	3
3.2	Web Applications VS Native Mobile Applications . . . . .	4

# 1 Requirements

## 1.1 Functional Requirements

1. The user should be able to search by geographic region and discover submitted plans for that region.
2. The user should be able to contribute routes.
  - 2.1. Only the creating user should be able to modify these routes
  - 2.2. The user should be able to decide on the visibility of this route (private or public)
3. The user should be able to interact socially with the route, including:
  - 3.1. Comment on routes
  - 3.2. Recommend alternative routes
  - 3.3. Share routes to external social media websites
4. Users should be able to create accounts and specify for example, name, age, email and location
5. There should be administrative users who have extra functionality available, including:
  - 5.1. Manage users
    - 5.1.1. Delete users
    - 5.1.2. Update users
    - 5.1.3. Create users
  - 5.2. Manage routes
    - 5.2.1. Delete routes
    - 5.2.2. Update routes
  - 5.3. Moderate comments
    - 5.3.1. Delete comments
    - 5.3.2. Update comments
  - 5.4. Make announcements
6. Users should be able to export their journeys
7. Users should be able to fork other user's public journeys (make a copy), and edit as they wish
8. There should be a journey editor component which allows the user's to create and specify their route
9. There should be a journey viewer component which visualizes a user's route

## 1.2 Non-Functional Requirements

TBD

## 2 Annotated Designs

### 2.1 Design One

### 2.2 Design Two

### 2.3 Design Three

Main pages of the app-

1. Landing Page

Journey viewers fills the majority of the page, search functionality above allowing users to find interesting journeys by geographic location

2. Account Creation Page

Allow users to create accounts, potentially through Google, twitter and facebook?

3. Login Page

Allow existing users to log in

4. Account Details Page

View and update account details, list their journeys, list their comments and updates of comments made on their journeys (and forks?)

5. Administration Page

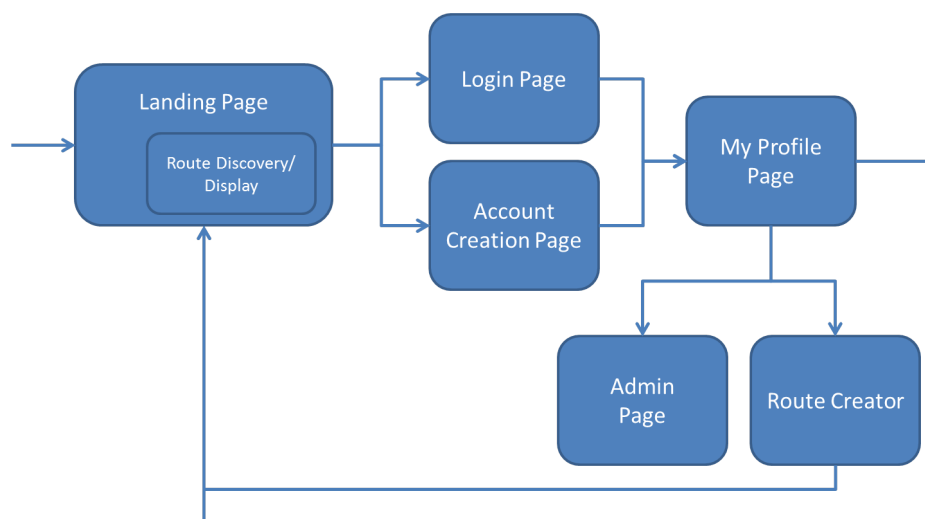
View real time and historic data for load levels, active users, journey creations, sessions, etc.

6. Route Creation Page

7. Route Discovery Page

Members only page similar to landing page but with social commentary

### 2.4 User Flow Diagram



## 3 Market Research

### 3.1 Analysis Of Similar Applications

Google Maps

## City Mapper

### Apple Maps

## 3.2 Web Applications VS Native Mobile Applications

### Web Applications

Web applications are easy to write and don't require knowledge of a specialized language, as the front end is in HTML, CSS and JavaScript, and the backend can be whichever language the programmer is most comfortable with. As they are hosted on the internet, there is no need for the user to download them, which is something that sometimes puts people off from using a service, this also means that if the application is updated, they don't need to install any updates, and can get straight back to using the application. Web applications don't require any approval before being distributed, as they don't need to be put on an app store, and as soon as they are up, they are available to everyone, regardless of their device (unlike mobile apps, which generally target one specific operating system). However, not being on an app store means it's more difficult to discover the application; native applications, are all visible within the market place, which is a lot smaller than the entirety of the world wide web. Another negative of web applications is that they require an internet connection to access (as they are web based), and they cannot fully utilise the native technology of the device they are on (unlike mobile apps which can access all the hardware of the phone they are on). However, web application advertisements are generally less intrusive than mobile ones, so running the application for free is much easier.

### Native Mobile Applications

Native mobile applications are good because they allow for greater exposure, as they have to be distributed through the app market place. This means there is a centralised place that your application will always be advertised on (although there is a negative that the application has to be submitted to the app store, and may require modifications to get it accepted, which could waste valuable time). This allows for multiple pricing models, a payment for downloading the app (which is less appealing for the user), or for free, with adverts (better, but sometimes the adverts can be very intrusive), or a free version, with the option to upgrade to remove adverts (best of both worlds). Native mobile applications can also be used offline, as they (generally) download all the content onto the user's phone, so it can be accessed regardless of internet connection (this however, is not really applicable to this project, because of the nature of it). This does mean, however, that the app needs to be downloaded (which is generally accepted nowadays, but could also lead to the app being ignored if the user, for instance, doesn't have enough space to download it), and any further updates to the application would need to be downloaded as well. While the application can utilise all the phone's native technology, it is only available for the platform that it was designed for (unless a tool to convert it is used, but this generally produces poor quality products).