|  |
| --- |
| Inf2C – Software Engineering. Coursework 1. Requirements Engineering |
| [Your Name]Tianyu “Tim” Wang (s1636723)October 11, 2017 |

Index

[Introduction of TourED 2](#_Toc495517482)

[Stakeholders 2](#_Toc495517483)

[Use case 1 – Tourist browses 3](#_Toc495517484)

[Use case 2 – User authors 5](#_Toc495517485)

[Use case 3 – Tourist follows 7](#_Toc495517486)

# Introduction of TourED

TourED is a mobile application for touring points of attraction in the City of Edinburgh. The majority of the tours will be on outdoors on foot and bicycle. An interactive on screen map will be provided to help the user find their bearings. TourED will come with a couple of pre-programmed tours with the option of users creating their own.

# Stakeholders

*The local government*. Tourism is an important contributor to GDP. According to an article from the Scottish Government's website, spending by tourists in Scotland generates £12 billion of economic activity. This suggests the Scottish Government may take an interest in TourED because the app could bring more tourists to Edinburgh.

*The local population* have a steak in TourED because tourists provide opportunity for the growth and development of small businesses. For example, in the Canongate area of Edinburgh there's Holyrood Palace, a common place of interest. An increase in people touring the area could provide residents of Canongate the opportunity to start a small business, e.g. a family owned restaurant.

*Tourists* are an obvious stakeholders of TourED, they are the end users of this application. They are using TourED to find good attractions, local food restaurants and souvenir shops. TourED will help them to find these sites and arrange a route for them to save time. Tourists are seeking an enriched travel experience in Edinburgh with the help of TourED.

*Owners of attractions* are also stakeholders of TourED, they want their sites be visited by tourists so they can sell tickets and make money. TourED have the ability to navigate tourists to different place, so they may want to create some tour on TourED to bring more tourists to their place.

# Use case 1 – Tourist browses

**Primary actor** Tourist

**Summary** Tourist wants to find a tour to follow on the system.

**Precondition** The available tours have loaded on the app.

**Trigger** Tourist is not following any tour at the moment.

**Guarantee** The tourist is on the tour.

**Main Success Scenario**

1. User opens tour app
2. App determines that the user is not following a tour
3. App determines the user’s region
4. App checks if there’s atleast 1 predefined tour in the region
5. Tours available are loaded onto the app

**Extension (2)**

1. App determines the user is following a tour
2. A message is delivered to the user stating “do you wish to stop your current tour?”
3. Follow on from point 3 in main success scenario.

**Extension (3)**

1. App cannot determine the user’s region
2. An error message is delivered to the user stating “Region cannot be identified”
3. Return out of main success scenario.

**Extension (4)**

1. App cannot find a predefined tour in the user’s region
2. An error message is delivered to the user “No available tours in region”
3. Return out of main success scenario.

**Notes**

* It should be considered what happens if the user changes region whilst browsing. E.g. user moves from EH1 to EH2 whilst browsing tours on their phone.

# Use case 2 – User authors

**Primary actor** User

**Summary** Author wants to define a tour for tourists to follow.

**Precondition** User is on the tour authoring interface.

**Trigger** User segues to tour authoring interface.

**Guarantee** The new tour is added to the list of tours in the app.

**Main Success Scenario**

1. User enters the name of the tour
2. User enters the tour’s region
3. User enters a sequence of waypoints and optional annotations.
4. User enters another set of optional annotations corresponding to a pair of waypoints; leg annotations.
5. User presses “publish tour”
6. Tour is added to the apps memory.

**Extension (3/4/5)**

1. User fails to enter the name of the tour, the tour’s region, the sequence of waypoints
2. App delivers a corresponding error message asking the user to enter the missed details.
3. Return to main success scenario.

**Extension (4)**

1. User enters an invalid region.
2. An error message is delivered to the user stating “Region cannot be identified”
3. Return back to the main success scenario.

**Extension (4)**

1. User enters an invalid waypoint that doesn’t exist.
2. An error message is delivered to the user “No available tours in region”
3. Return out of main success scenario.

**Notes**

* A definition of a valid annotation should be considered because some people might enter rude or nonsensical things. For example, a filter for swear words could be added.
* Since the tours are not stored on a server, we might consider a maximum number of tours the app can store locally.

# Use case 3 – Tourist follows

**Primary actor** Tourist

**Summary** the tourist want to have a trip navigated by TourED

**Precondition** tourist is browsing tours

**Trigger** tourist select a tour.

**Success Guarantee** tourist get the end way point of the tour

**Failure Guarantee** tourist get lost and go to somewhere else

**Main Success Scenario**

1. TourED display the sequence of *waypoints* linked together on the map with a short description.
2. Tourist click 'start tour'
3. TourED show the position, distance of the start waypoint with the closest route to it on map
4. Tourist follow the route
5. Tourist arrive at the waypoint
6. TourED show annotations of this waypoint if there is one
7. Tourist click 'next'
8. (if this waypoint is not end waypoint) TourED show the position, distance of next waypoint with the leg to that waypoint and a small annotation of the leg if there is one
9. Tourist follow the leg
10. Tourist arrive at the next waypoint
11. repeat from 6 until the current waypoint is the end of the waypoint
12. TourED show annotations of the end waypoint
13. Tourist click 'exist'

**Extension 1**

1. Tourist didn't follow the route and have not gone far
2. TourED send notification with new route on it (device will shock or make sound)
3. Tourist see the notification
4. Back to 4 in main scenario

**Extension 2**

1. Tourist didn't follow the leg and have not gone far
2. TourED send notification with route back to leg (device will shock or make sound)
3. Tourist see the notify
4. Tourist go back to leg
5. Back to 9 in main scenario

**Extension 3**

1. Tourist didn't follow the route and have gone too far
2. TourED cancel the tour and send notification (device will shock or make sound)