# CS243 Software Engineering Course Project

Team No. 3

Project No. 4

## Immersive Virtual Tour

Software Requirements Specification Document

Abhishek Suryavanshi (160101009)
Ameya Daigavane (160101082)
Nitesh Jindal (160101084)

# Content

# 1 Introduction

This Software Requirement Specification (SRS) Document describes an Android application that provides an immersive virtual tour of the Department of Computer Science and Engineering, IIT Guwahati.

## 1.1 Purpose

The SRS describes the functional and non-functional requirements of the Virtual Tour Application. The intended audience for this SRS is Professor Samit Bhattacharya, who is in charge of the CS-243: Software Engineering course.

## 1.2 Scope

The software that this SRS specifies is the Immersive Virtual Tour (IVT) Application. This is an Android application that will allow a Virtual-Reality (VR) based three-dimensional tour of the Department of Computer Science and Engineering, IIT Guwahati - through a external VR headset.

## 1.3 Definitions, Acronyms, and Abbreviations

User : Person interacting with the application
SRS : Software Requirements Specifications
IVT :  Immersive Virtual Tour
VR : Virtual Reality
SDK : Software Development Kit
NDK : Native Development Kit
JDK : Java Development Kit
GPU : Graphics Processing Unit

APK : Android Package
Department : Department of Computer Science and Engineering, IIT Guwahati

## 1.4 References

“IEEE Recommended Practice for Software Requirements Specifications”, IEEE Software Engineering Standards Committee, IEEE Std 830-1998, October 20, 1998.

## 1.5 Overview

The remaining part of the SRS contains:

a. The Overall Description and Functioning of the Software
b. Specific Requirements:
    i. Functional - defining the fundamental actions that the software incorporates in accepting and processing the inputs and corresponding outputs.
    ii. Non-Functional - software system attributes that are used to judge the operation of the system.

# 2. The Overall Description

The following subsections serve as a background for the functional requirements, defined in Section 3.

## 2.1 Product Perspective

This product is a stand-alone Android application that can be launched after installing via a standard freely-distributed APK file.
The product requires the device's gyroscope and accelerometer to function - these are accessed through the Android System Interface after requesting permissions from the user.
Without these, the application will not function. A VR headset is required to see the 3D representation of the Department. No other special hardware or software interfaces are required.

### 2.1.1 System Interfaces

On launching the application will require the Permissions API on the Android System to access the gyroscope and accelerometer data, the VR display and the audio playback device. If permissions are granted, system APIs are required to read the sensor data.

### 2.1.2 User Interfaces

a. The product will be accessible to any user on a compatible Android device and VR headset.
b. The user interacts with the virtual world through a visual interface displayed through the VR headset.
c. To change certain accessibility settings and audio/resolution preferences, a menu-based interface is provided which can be accessed through the VR headset buttons.

### 2.1.4 Memory constraints

a. The app requires a minimum of 1 GB of primary memory (RAM) and 512 MB of secondary memory for installation and execution.

## 2.2 Product Functions

Functions included in the final product will be as follows :-

1. Virtual World Generation
2. User State Modifier
3. Sound Generator
4. Virtual World Interaction Handler
5. Global Properties Modifier

## 2.3 Assumptions and Dependencies

**Assumptions:**

1. Device uses Android 2.2 or higher / API level 8 or higher.
2. A functioning VR headset is required.
3. Device must have a working gyroscope and accelerometer.
4. Device has Qualcomm Adreno 506 or higher GPU.
5. Device has ARM Cortex-A53 or higher CPU.

**Dependencies:**

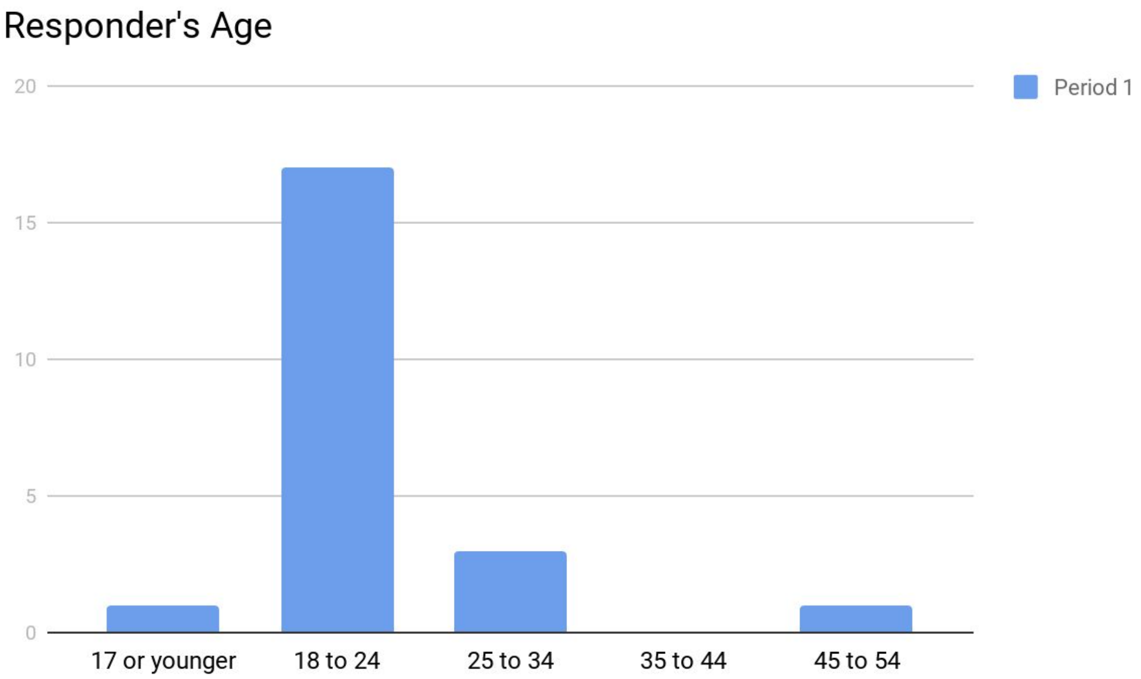
1. Graphic drivers installed device.
2. Audio drivers of the device.

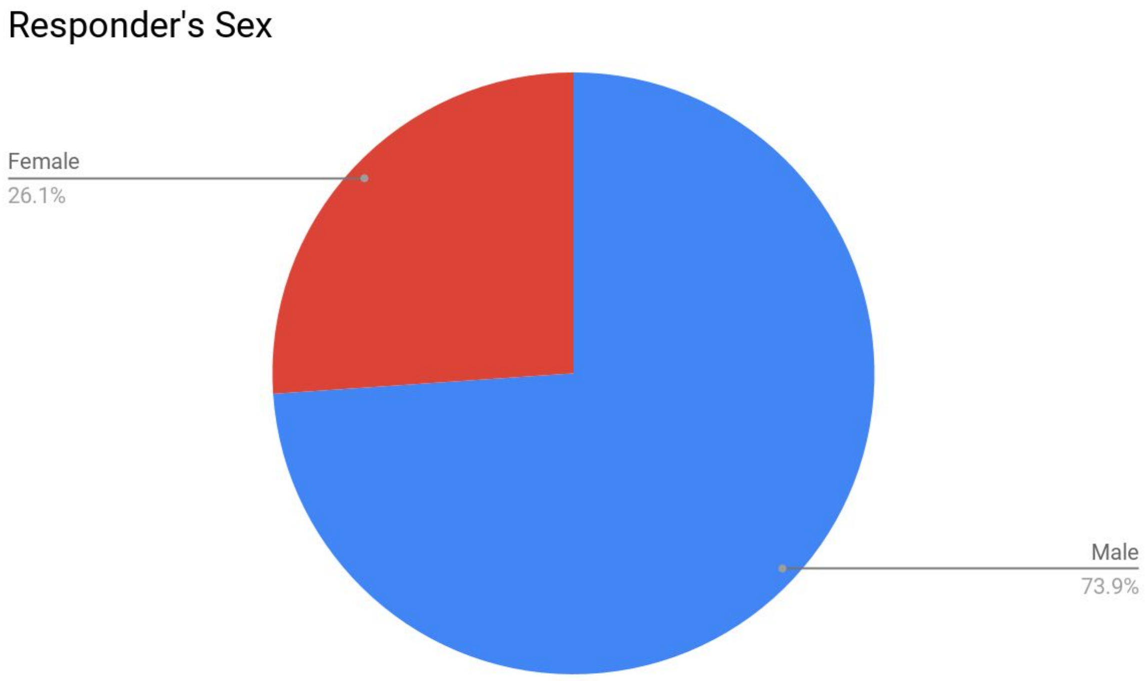
## 2.4 User Characteristics

The intended users for the product will have the following characteristics:
1. No visual disabilities - the ability to perceive the virtual world should not be hindered.
2. No health conditions such as excessive light-sensitivity that may trigger seizures.
3. Is able to understand English.
4. Has basic motor skills in order to use the interface.

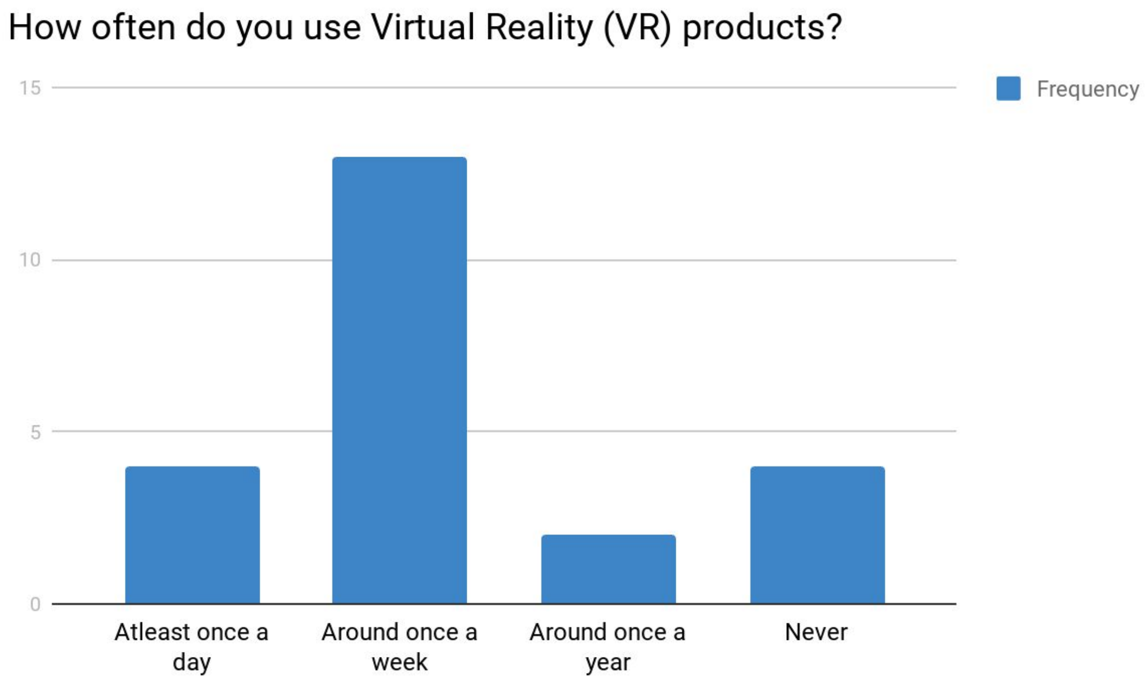
## 2.5 Contextual Inquiry

In order to estimate user requirements, we made a Typeform survey (link: https://ameyadaigavane.typeform.com/to/oTrWna) with 10 questions. We got a total of around 25 responses. Below, we analyse the answers and justifying the majority of them.

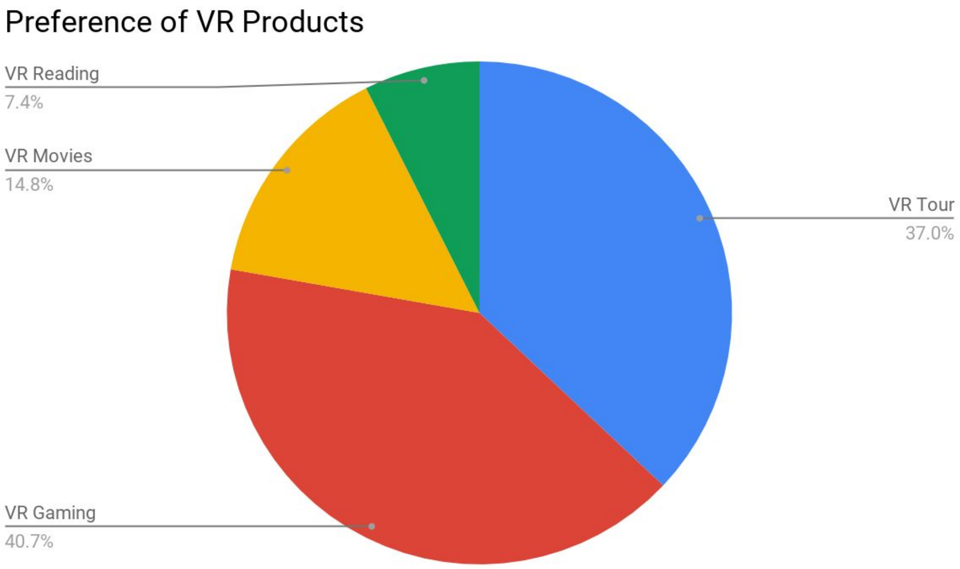




Our respondent demographic is skewed towards the young male adult population. However, we can argue that this is not a significant deviation from the main users of the IVT application - the students of IIT Guwahati, and hence the responses will be mostly representative of the overall target audience.

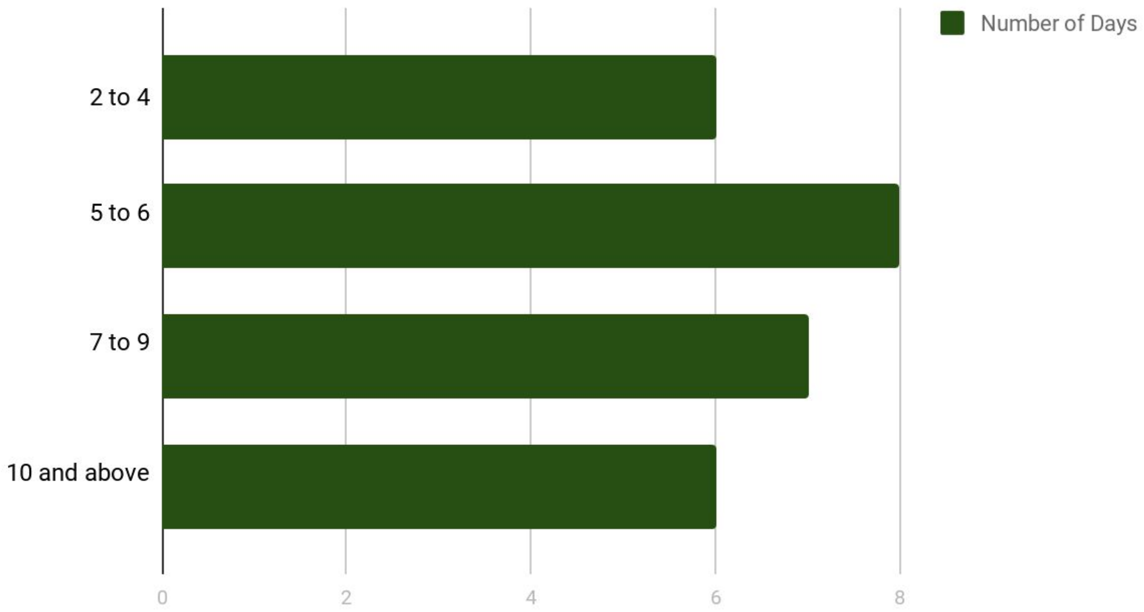


Most of our respondents said that they use a VR product at least once a week. Considering that most respondents are young adults, and the advent of virtual gaming, this is not very surprising. However, there is greater market potential for VR - we still have around a quarter of the target population using VR sporadically or not at all.



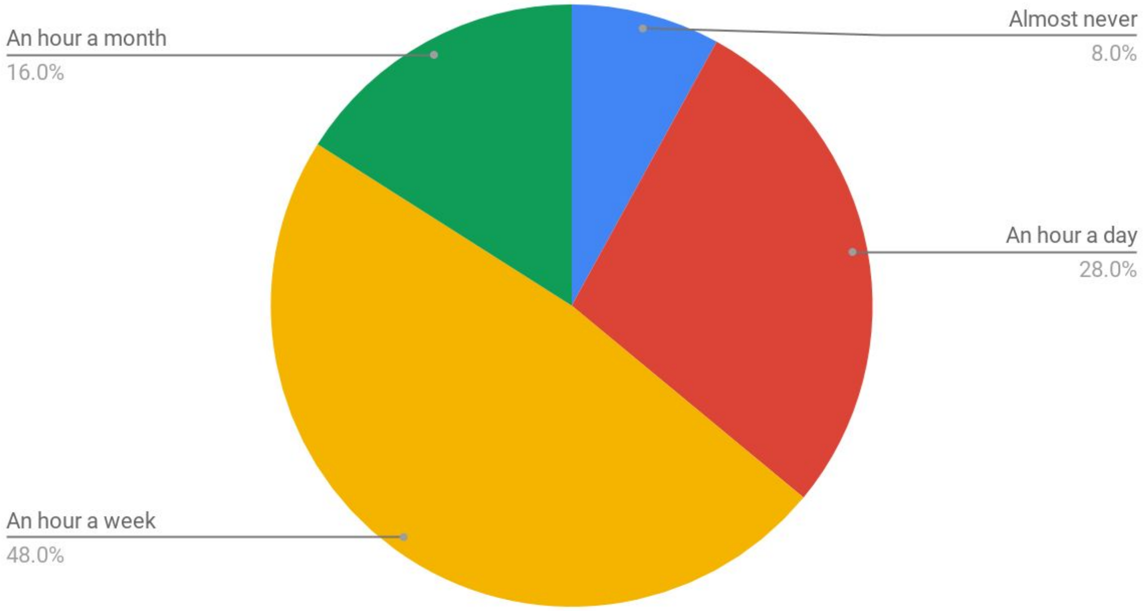
The craze behind VR Gaming is evident. However, VR Touring - visiting places virtually - is also pretty popular - as the popularity of Google's StreetView worldwide shows.

Average number of days in a year spent travelling to new places

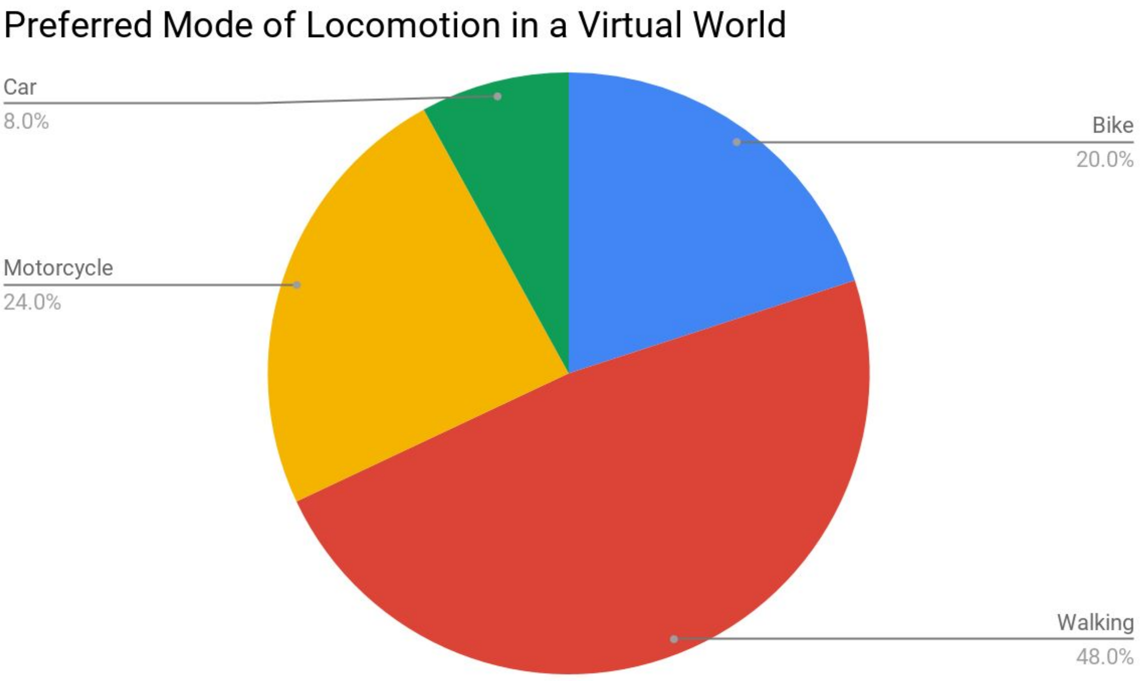


It is clear that the average student does not do too much travelling. Virtual Reality tours have the ability to bring the experience of travelling to an undiscovered location to many.

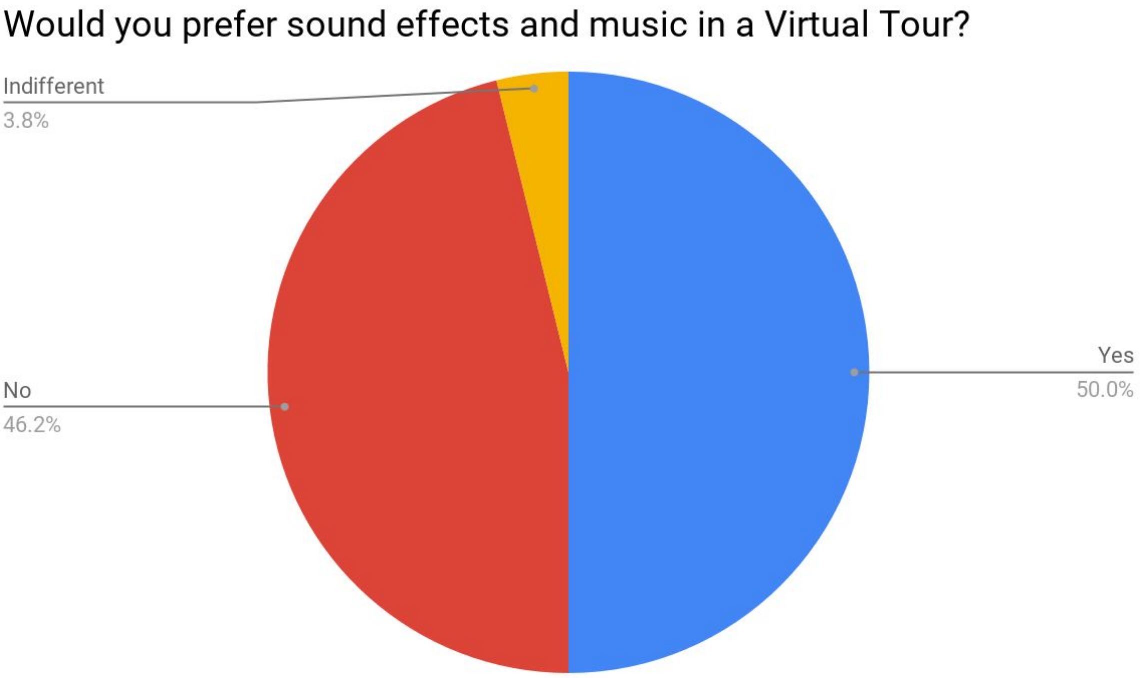
How often would you use an Immersive Virtual Reality Tour app?



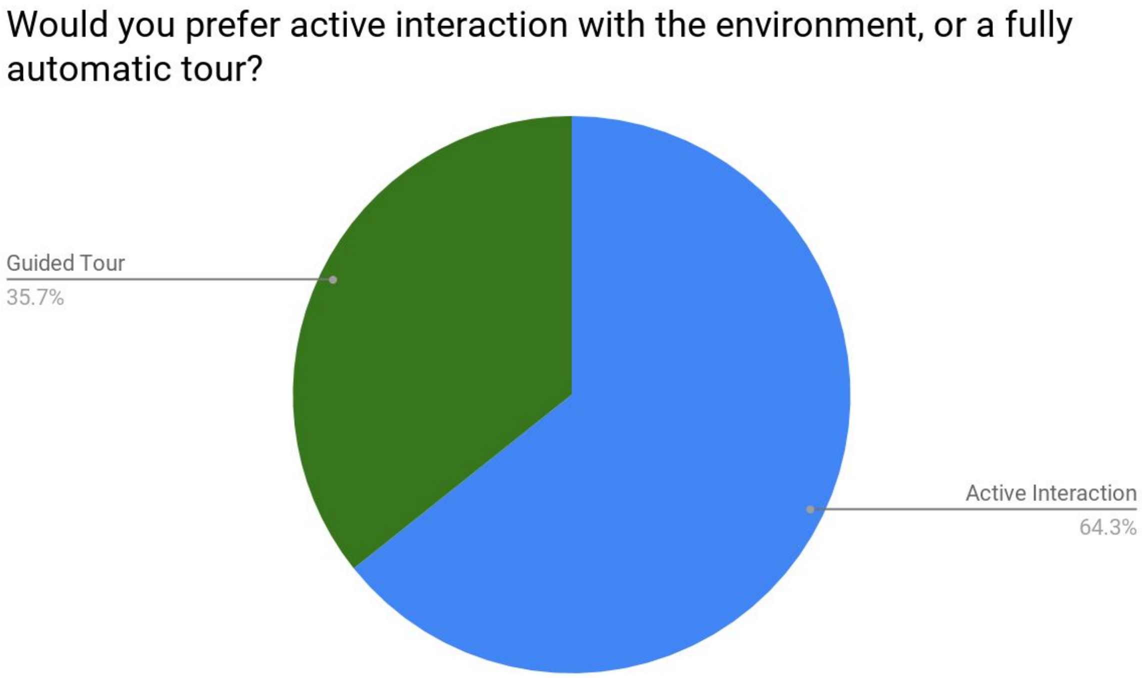
Most users would not use a VR Tour app extremely frequently, but an hour a week is sufficient enough to indicate a sizeable market.



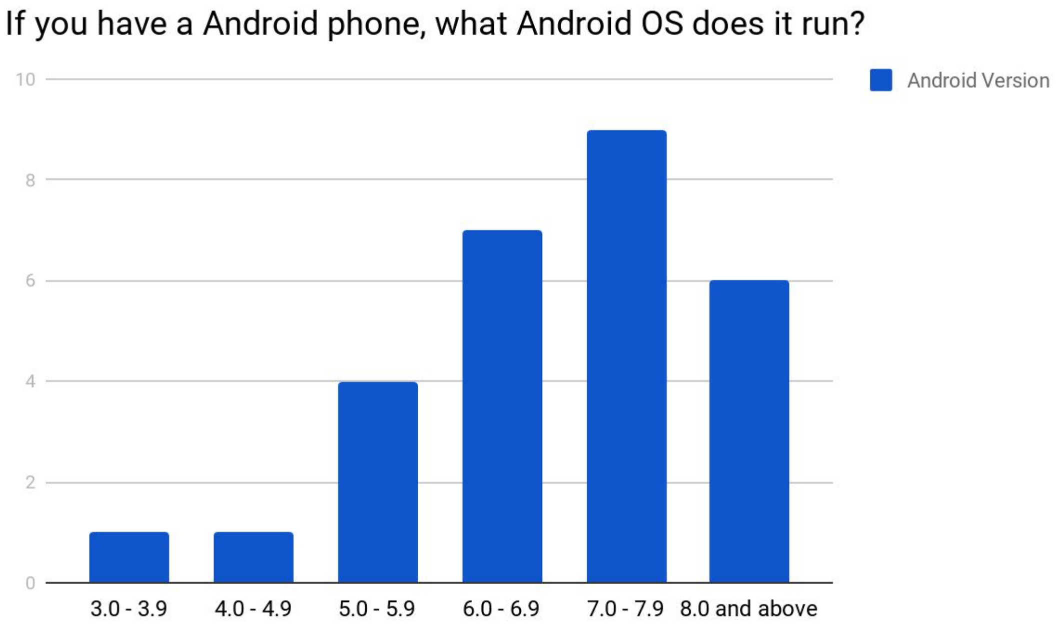
It is natural to expect many people to go walking in a virtual world - enabling them to interact much more freely with the environment, at the cost of pace.



Music and sound effects preferences were split across our responders. This suggests that a audio enable/disable option is a good idea when developing the IVT application.



Most responders wanted to actively interact with the environment, instead of sitting back and let the application show them around. However, having a guided tour option may help satisfy those who prefer to relax in the virtual world.



To ensure compatibility issues are minimized for the development team, we asked respondents for the Android OS version that their phones ran. Note that most VR displays and applications won't be supported for Android OS versions less than 5.0. However, this should not cause an issue with most of our target audience.

# 3. Specific Requirements

## 3.1 External Interface Requirements

This section describes all required inputs, outputs and interfaces that the functional requirements will utilize.

**Inputs**

- World Specification File - This is a file that contains all the physical descriptions of everything in the virtual world, including items and characters, present in the application folders in memory.
- Access to Device Internal and External Memory
- Gyroscope Data
- Accelerometer Data

**Outputs**

- Display onto the VR Headset
- Vibrational feedback in the VR Headset
- Audio playback through playback device

### 3.1.1 Hardware Interfaces

The application has no designated hardware so there are no direct hardware interfaces. The vibration feedback for the VR headset is handled by the headset drivers installed on the system.

### 3.1.2 Software Interfaces

The mobile application communicates with the gyroscope and accelerometer of the phone.The communication between the them and the mobile application consists of only reading operations.

## 3.2 Functional Requirements

### 3.2.1 Virtual World Generation

**Input:** World Specification File, and the user starts the tour
**Output:** Representation of the 3D world virtually

**Description:**
On initiating the tour, the 3D world is generated as a modifiable object, described by physical features (structure, terrain, weather, acoustics) which can be observed and interacted upon by the user, according to the World Specification File.

#### 3.2.1.1 Terrain and Structure Creation

**Input:** World Specification File
**Output:** Display of the three-dimensional structure of the world.

**Description:**
Generates the structure and terrain of the 3D world on which the user will navigate. This also sets the boundaries that the user will encounter on the tour.

#### 3.2.1.2 Features-of-Interest Notification

**Input:** World Specification File, and when user enters a region of interest
**Output:** Notification text with description of region popup

**Description:**
Certain rooms and areas will be labelled with a name and description tag. When the user enters these areas, descriptive text about the area will popup on the display.

#### 3.2.1.3 Weather Generation

**Input:** World Specification File
**Output:** Display of weather conditions along with associated audio playback.

**Description:**
Weather characteristics - sunny, cloudy, rainy - are shown to the user along with associated audio output.
**Dependencies:** 3.2.3.1

#### 3.2.1.4 Characters Generation

**Input:** World Specification File
**Output:** Addition of Characters with dialogue and physical models to the virtual world.

**Description:**
The virtual world will contain virtual characters - such as people and animals - that move around, can be interacted with and talked to. For each real virtual character specified in the World Specifications File, this feature adds it to the virtual world.
**Dependencies:** 3.2.3.1 and 3.2.3.2

#### 3.2.1.5 Item Generation

**Input:** World Specification File
**Output:** Addition of Items with physical models to the virtual world.

**Description:**
The virtual world will also contain inanimate objects - such as boxes and computers - that do not have any independent motion but can be interacted with, by the user. For each item specified in the World

Specifications File, this feature adds it to the virtual world.
**Dependencies:** 3.2.3.1 and 3.2.3.2

### 3.2.2 User State Modification

**Input:** Gyroscope and accelerometer readings, user current coordinates and direction.
**Output:** Change in user state properties (position, field-of-view, direction).

**Description:**
Depending on the actions performed by the user in real life, either the position, direction, or field-of-view properties of the user in the virtual world (generated above) are modified.

#### 3.2.2.1 User Movement Handler

**Input:** Current coordinates of the user, Gyroscope and Accelerometer Data.
**Output:** Updated coordinates of the user.

**Description:**
If the gyroscope and accelerometer readings change due to the user's actions, indicating translational movement, then this feature updates the user's coordinates, relative to the world.

#### 3.2.2.2 User Direction Handler

**Input:** Current direction vector of the User, Gyroscope and Accelerometer Data.
**Output:** Updated direction vector of the User.

**Description:**
If the gyroscope and accelerometer readings change due to the user's actions, indicating pitch, roll or yaw

movement, then this feature updates the user's direction vector and their field-of-view in the world.

### 3.2.3 Sound Generation

**Input:** Current coordinates and direction of the user, World Specification File
**Output:** Audio associated with different objects and locations in the world

**Description:**
For a greater degree of immersion, sound will be outputted through the audio playback device, depending on the user's current coordinates and motion.

#### 3.2.3.1 Static Sounds Creation

**Input:** World Specification File
**Output:** Audio Playback
**Description:**
Certain sound sources will be active at different locations in the 3D world irrespective of the user's current position - for example, machinery and weather sounds. The intensity of the sound will however, depend on the distance between the source and the user.

#### 3.2.3.2 Dynamic Sounds Creation

**Input:** Current coordinates and direction of the user, World Specification File
**Output:** Audio Playback

**Description:**
Produces or modifies sounds - such as footsteps and hitting a wall - depending on the user's interaction with the world.

### 3.2.4 Virtual World Interaction

**Input:** User’s current coordinates and direction, World Specification File

**Output:** Display changes and audio playback.

**Description:**

While touring the virtual world, the user will be allowed to interact with certain objects.

#### 3.2.4.1 User-Boundary Interaction

**Input:** User’s current coordinates and direction, when the user hits a boundary.

**Output:** Stoppage of user motion and audio playback

**Description:**

This feature stops the user’s movement when the user hits a boundary in the virtual world, and also produces some audio indicating contact.

**Dependencies:** 3.2.1.1 and 3.2.3.2

#### 3.2.4.2 User-Character Interaction

**Input:** User initiates interaction with a Character

**Output:** Modification in characters in the 3d world, Audio output.

**Description:**

When the user wants to interact with the Characters in the world, this function takes the interaction requests and produces the corresponding graphic or audio response.

**Dependencies:** 3.2.1.4 and 3.2.3.2

#### 3.2.4.3 User-Item interaction

**Input:** User’s current coordinates and direction, Interaction flag (Boolean).

**Output:** Modification in the objects in the 3D world, Audio output.

**Description:**

When the user wants to interact with the inanimate objects in the world this function takes the interaction requests and produces the corresponding graphic or audio response.

**Dependencies:** 3.2.1.5 and 3.2.3.2

### 3.2.5 Global Properties Modifier

**Input:** Modify Property Request flag (Boolean), New Value of Property (Integer)

**Output:** Change in Global Properties Object

**Description:**

When the user wants to modify some property of the world (where the property is sourced from a list of modifications, such as volume, vibration levels, tour mode, audio disable/enable, and so on), the property is modified to the new value, changing the Global Properties Object.

#### 3.2.5.1 Guided Tour or Free Roam

**Input:** Guided Free Roaming flag (Boolean).

**Output:** Enable user to switch between guided tour and free roam.

**Description:**

In the Guided Tour mode, user movement and direction handlers are disabled, neglecting user inputs

to give an automatic tour. In the Free Roam mode, these are enabled, allowing user freedom to navigate.
**Dependencies:** 3.2.2.1 and 3.2.2.2

#### 3.2.5.2 Settings

**Input:** Modify Property Request flag (Boolean), New Value of Property (Integer)
**Output:** Modification of certain physical property

**Description:**
Handles properties such as changes in volume, resolution, vibration feedback or font sizing.

##### 3.2.5.2.1 Modify Resolution

**Input:** Modify Resolution flag (Boolean), New Value for Resolution (Integer : range 0 to 5)
**Output:** Change in resolution of the game.

**Description:**
This function will change the resolution of the game on user demand.

##### 3.2.5.2.2 Modify Volume

**Input:** Modify Volume flag (Boolean), New Value of Volume (Integer: range 0 to 100)
**Output:** Modifies the audio volume to the new resolution

##### 3.2.5.2.3 Modify Vibration Level

**Input:** Modify Vibration flag (Boolean), New Value of Vibration Level (Integer)
**Output:** Enables game to give vibration feedback on user demand.

#### 3.2.5.1.4 Modify Font Size

**Input:** Modify Font Size flag (Boolean), New Value of Font Size (Integer)
**Output:** Modification of Font Size Property in Global Properties Object

## 3.3 Software System Attributes

This section includes all the non functional requirements for the software.

### 3.3.1 Reliability

The software is supposed to work smoothly without any failure. The application will stop running if sent to run in the background, or the phone is accidentally powered off.

### 3.3.2 Availability

The application will restart from scratch once exited. The user won't be able to use any other application in the mobile while using this software.

### 3.3.3 Security

The software does not collect or share any personal data when running. No network communications occur due to execution of the app.

### 3.3.4 Maintainability

The system will be updated with software patches through the Google Play Store. Updates can be downloaded through the standard Android interface.

### 3.3.5 Portability

The software can be used on any Android phone satisfying the minimum hardware/software dependencies as specified in this SRS document previously. Installation of this application can be done through the standard Android File Manager, and this application can be shared through an APK file between devices.