Software Design Document

For
Virtual Tour of
Department of Computer Science and Engineering

Version 1.0.1

Prepared by
Group 27
Abhinav Hinger-160101004
Apurva N. Saraogi-160101013
Daman Tekchandani-160101024

Date
15th March 2018

Supervised By
Prof. Samit Bhattacharya

Contents

1 Introduction

This Software Design Document is a document to provide documentation which will be used to aid in software development by providing the details for how the software should be built. Within this Software Design Document are narrative and graphical documentation of the software design for the project including use case models, sequence diagrams, class diagrams and other supporting requirement information.

1.1 Purpose

The purpose of this document is to give a detailed description of the design for the Virtual Tour of Department of Computer Science and Engineering. This includes use case models,sequence diagrams, class diagrams and other supporting information. It will illustrate the complete declaration for the development of system along with the system constraints. This document is primarily intended to as a reference for developing the first version of the system for the development team.

1.2 Scope of Project

The VR App when completed can be used by all users who know how to operate an Android application and a VR headset to get an immersive tour of the Department.This app can also help in swiftly navigating through the complex CSE department. Most of the gestures to roam around in the virtual world are kept intuitive which makes it easy to operate.

1.3 Document Conventions

Term	Definition
Tourist User	User who wants to tour the Department and is familiar with the working of a smartphone and a VR headset.
VR Headset	A **virtual reality headset** is a heads-up display (HUD) that allows users to interact with simulated environments and experience a first-person view (FPV).
Accelerometer	Sensor that helps determining movement of Smartphone.
Gyroscope	Sensor that helps determine the orientation of Smartphone
Sensors	Collectively refers to the Accelerometer, Gyroscope
UI	Full Form : User Interface. Interface which interacts with user

2 Design Overview

2.1 Introduction

The Design Overview is section to introduce and give a brief overview of the design. The System Architecture is a way to give the overall view of a system and to place it into context with external systems. This allows for the reader to get brief a summary before proceeding into the details of the design.

2.2 Background

This Application provides user with a Virtual Tour of the Computer Science and Engineering Department of IIT Guwahati. User just need to have knowledge of using a VR headset. Movements in Virtual Tour are kept intuitive so that it will be user-friendly. This Application will help people to get insight of the department.

2.3 System Architecture

User Starts the Application and is positioned at the entrance of CSE Department. Using Head Movements he navigates the augmented user throughout the department. 3D View of the user keeps updating according to user's current position and his view orientation. Head Movements are tracked in real time by built in sensors in the smartphone(accelerometer/gyroscope).

User can also view the details of a particular section of the Department i.e. Office of a Professor, Lab, Library, etc. This is done by using Gaze Clicking the required Department. Gaze Clicking is keeping the augmented user's center point of view fixated on the door of a room for a fix period of time. Probability of error is drastically reduced by checking whether the room is in vicinity of the user or not.

2.4 System Interfaces

2.4.1 Hardware Interfaces

1. The final version of the software can be used by any smartphone with at least:
    - Android 4.4+ (KitKat)
    - Processing Power: Dual Core 1.2GHz
    - Accelerometer Sensor

2. VR headset compatible with Google Cardboard SDK

2.4.2 Communication Interfaces

Map will be Offline without the need for a Server side application.Environmental sensor data will be sent by the Android as sensor information packages through Java's ObjectOutputStream.

2.4.3 Software Interfaces

This Application will be created using Unity, Blender and Android Studio. Database will be implemented by Unity itself.

2.5 Assumptions and Dependencies

1. OS used on the device :- Android 4.4 (KitKat) or higher
2. Device must have a minimum of 500 MB storage space for storing the application and also some minimal data generated whilst operation.
3. Device must have a minimum 2GB RAM and 1.2GHz Processing Power.
4. Device must have support for Virtual Reality Apps
5. Must have ARM architecture based hardware design

3 Use Cases

3.1 Actors

3.1.1 Tourist User

Information: Tourist is a User who wants to tour the Department and is familiar with the working of a smartphone and a VR headset.

3.2 List of Use cases

3.2.1 Start Virtual Tour
3.2.2 Navigate
3.2.3 View Details
3.2.4 Exit Tour

3.3 Use Case Diagram

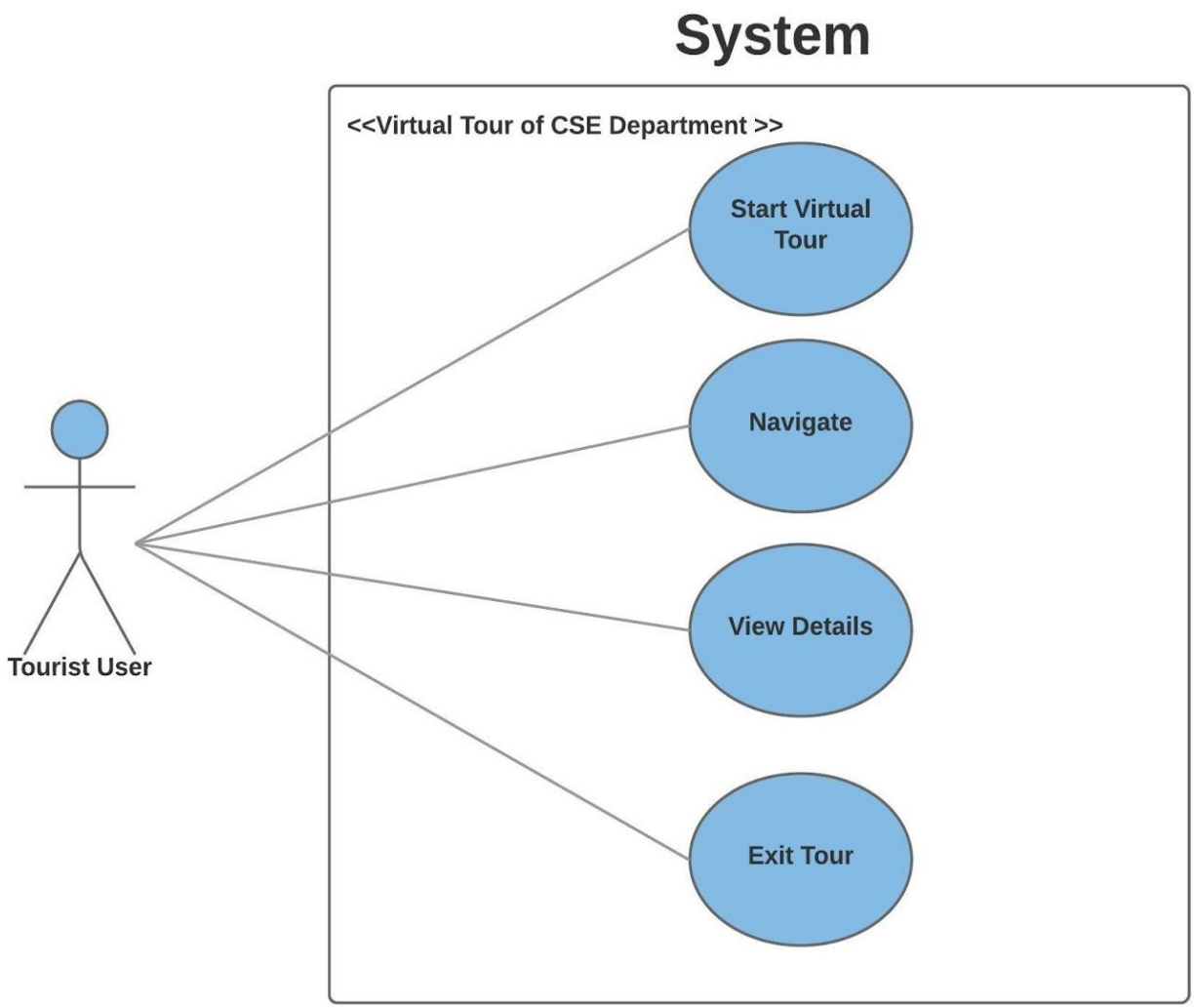


Figure 3.1 Use Case Diagram

3.4 Use Cases

Actor : Tourist User

U1 : Start Virtual Tour

Using this use case,tourist user can start taking the Immersive tour of the Department.

Scenario 1 : Mainline Sequence

1.User : Selects START TOUR button on home screen
2.System : Prompts to input user Name and Age
3.User : Enters the required details
4.System :Prompts successful profile creation and starts the virtual tour by loading the 3D Map data and placing the User at the entrance of the department

Scenario 2 : at step 4 of mainline sequence

1. System : displays the message that some input information has not been entered. The system displays a prompt to enter the missing value

U2 : Navigate

The use case records the external input regarding navigation of the user and makes changes to the 3D location and surroundings.

Precondition : User has started the tour and is in Virtual Environment

Scenario 1 : Mainline Sequence :

1.User : Moves his/her Head in one of the following ways:
    1.Forward
    2.Backward
    3.Left
    4.Right
    5.Look Up/Down
    6.Turn Left/Right

2.System : Reads the direction of the User and Updates 3D Surroundings of the Virtual User .

Scenario 2 : At step 2 of mainline sequence

2.System : Recognises that User cannot move in that direction.No change in 3D Space.

U3: View Details

Precondition : User is in Virtual Environment and in vicinity of an object which has description preassigned (Labs or Professors' information)

Scenario 1 : Mainline Sequence
1.User : Selects VIEW DETAILS button using hand held controllers or Gaze
2.System : PopUp Description or information on the corresponding object.

Scenario 2 : in step 2 of mainline sequence
1.System : Recognises no description is assigned to that Section of department(Professor or Labs etc.). No PopUp .

U4: Exit Tour

This use case is used to Exit from the application.

Precondition : User has removed VR but Application is on.

Scenario 1 : Mainline Sequence
1.User : Selects "EXIT"
2.System : Asks the User if he/she is sure to exit the Application.
3.User : Selects Yes or No
4.System : Does one of 2 things depending on the answer
    Yes : Takes the user out of the Application
    No : Stays in the Virtual Tour.

4 Object Oriented Model

4.1 Domain Modelling

4.1.1 Boundary Objects

1. 3D Visualisation
2. Room

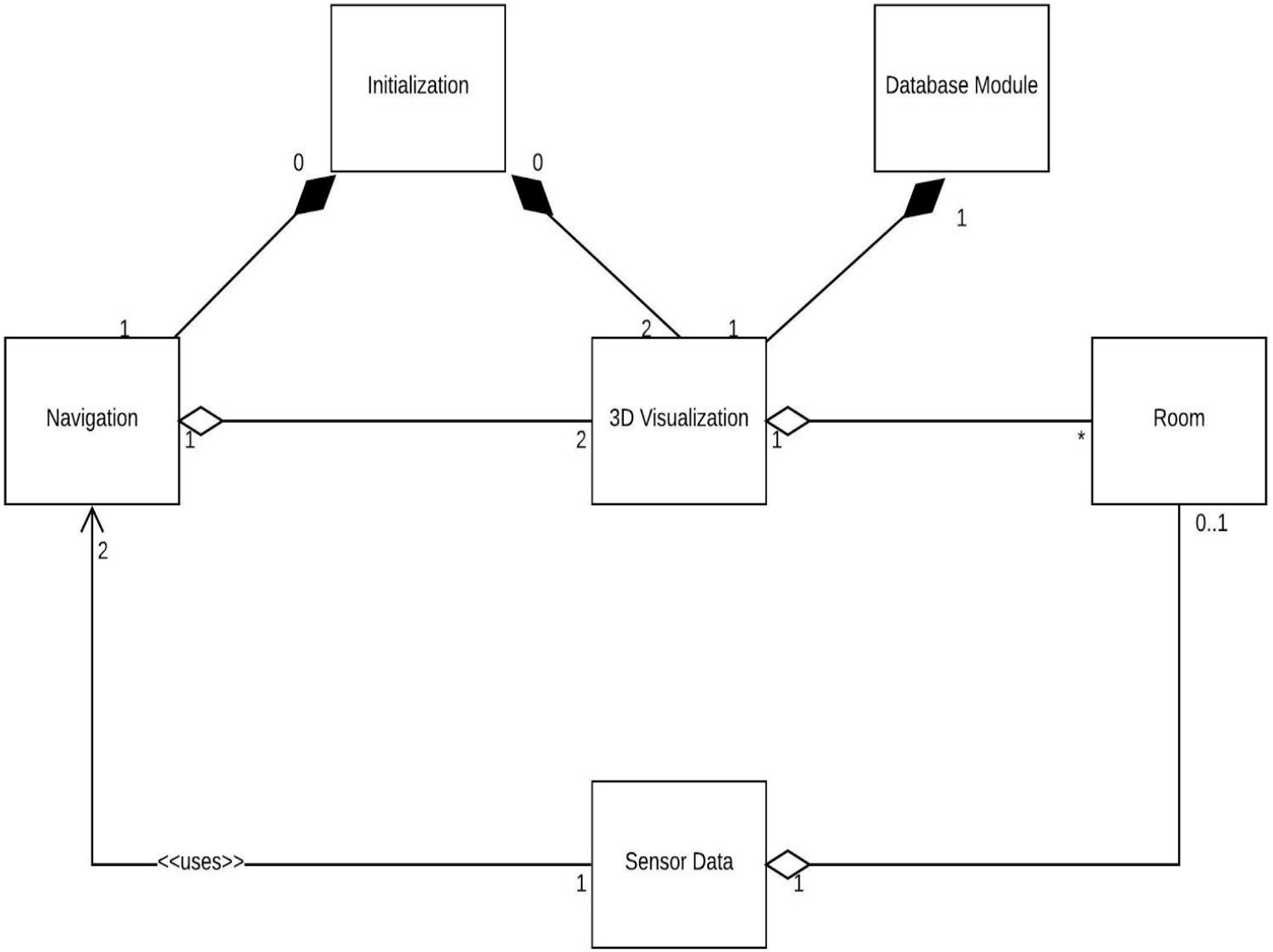
4.1.2 Controller Objects

1.Initialisation
2.Navigation

4.1.3 Entity Objects

1.Database
2.Sensor Data

4.2 Class Diagram


Initialization
Database Module
0
0
1
1
2
1
Navigation
3D Visualization
Room
1
2
1
*
2
0..1
<<uses>>
Sensor Data
1
1

4.3 Class Description

4.3.1 Boundary Objects

4.3.1.1 Room

Description :
Consists of the rooms i.e Offices and Labs that the User interacts with

Attributes:
1.public integer RoomID
Description : Unique ID of the selected room
2.public float <tuple<X,Y,Z>> RoomPosition
Description : Position of the Door of the room or Centre of Position
3.public string RoomDetails
Description : Stores the details which need to be popped up upon interaction

Methods:
1.public string returnDetails

1. Input:RoomID
2. Called by : Upon interaction by user
3. Output : String containing room details like professor details or lab functions which is popped up.

2.public checkInteraction

1. Input : Sensor Data
2. Output : RoomID
3. Description : This function constantly checks for user interaction with any of the links. On any interaction, it returns the Room ID of the selected link.

3.public popupDisplay

1. Input : Details of the RoomID
2. Output: Pop up the details of that Office or Lab
3. Description : This pops up the details of the object passed

4.3.1.2 3D Visualisation

Description :
Sets up the display of the Augmented User i.e 3D Visualisation using 3D Space Objects and Details

Attributes:
1.private 3DObject 3dSpace
Description : Three dimensional models making the view of the augmented user.
2.public float <tuple<X,Y,Z>> CoordinateVars
Description : Current Position of the Augmented User
3.public float <tuple<X,Y,Z>> CameraOrientation
Description : Current orientation of camera/view of the augmented user.
4.public Room listroom[]
Description : List the Important rooms (Office/Labs)in the department for which Details can be given
5.public details RoomDetails
Description : Stores the details of selected room(Office/Lab)

Methods:

1. public void build3DView
    1. Input: 3dSpace,CoordinateVars,CameraOrientation,listroom[]
    2. Called by : Whenever change in position of Augmented User
    3. Output : Void (just builds 3D View around the user)
    4. Description : Build 3D surroundings according to the position of the augmented user. Since the surroundings change whenever the user moves, this function is needed to constantly calculate the 3D surrounding setup for each and every position and orientation of the user's view.

2. public checkNearbyRoom
    1. Input : CoordinateVars,listRoom[]
    2. Output : Rooms close to the user
    3. Description : Filter a subset from the list of rooms which are close enough to the coordinate variables. The position property of the room object is used for this calculation.

4.3.2 Controller Objects

4.3.2.1 Initialisation

Description:
Call the database entity and retrieve data for making 3D visualization and room details. Initialize the user coordinate variables in the 3D space.

Methods:

1. public call3DInit
    1. Input : 3D space , listRooms[]
    2. Called by : Whenever user selects to start Tour
    3. Output: 3D Visualisation of the Department
    4. Description : Invoke 3d visualization entity passing the selected 3D object.

2.public navigationInit()

Initialize the navigation module (create an object).

3.public initUser

Initialise the user coordinates to the entrance of CSE Department and Camera orientation appropriately.

4.3.2.2 Navigation

Description: Change the coordinate variable of the augmented user in case of head movement. Also verify that the augmented user is within the boundaries of the Department.

Attributes:

1.public float <tuple<X,Y,Z>> CoordinateVars
Description : Current Position of the Augmented User
2.public float <tuple<X,Y,Z>> CameraOrientation
Description : Current orientation of camera/view of the augmented user.
3.public float <tuple<X,Y,Z>> Boundaries[]
Description: A list of boundary coordinates of the Department

Methods:

1.public moveLeft()
    1. Input : Sensor Data , CoordinateVars
    2. Called by : On the event of change in Sensor Data matrix due to head movement
    3. Output : New CoordinateVars of Augmented user
    4. Description : Shifts the augmented user left with respect to current line of sight hence changing the coordinate variables.Also checks the boundaries so that user does not cross the Department boundaries

2.public moveRight()
Similar as (1)

3.public moveFront()
Similar as (1)

4.public moveBack()
Similar as (1)

5.public rotateLeft()

1. Input : Sensor Data , CameraOrientation
2. Called by : On the event of change in Sensor Data matrix due to head movement
3. Output : New CameraOrientation of Augmented user
4. Description : Change the augmented user's line of sight/ Camera Orientation in the 3D visual certain degrees left about the vertical axis.

6.public rotateRight()
Similar as (5)

7.public pitchUp()
Similar as (5)

8.public pitchDown()
Similar as (5)

9.public checkValidMove

1. Input : CoordinateVars
2. Called by : Whenever there is change in Coordinates of the Augmented User
3. Output : Bool(Valid or Invalid)
4. Description: Validate whether or not the input coordinates is within the boundary[] list. If not, return error saying move is invalid.

4.3.3 Entity Objects

4.3.3.1 SensorData

Description: The sensor data recordings of Accelerometer/Gyroscope

Attributes
1. public float accelGravityData[3]
Description: The Accelerometer reading
2. public float rotationMatrix[16]
Description: The Gyroscope reading

4.3.3.2 Database Module

Description : This allows the app to retrieve information from the database as and when required by other entities.

Attributes :
1. public float <tuple<X,Y,Z>> Boundaries[ ]
Description: A list of boundary coordinates of the Department

Method :
1. connectDB() (CONSTRUCTOR)
    1. Input: Initial Coordinates
    2. Called By : 3D Visualization
    3. Output : Surrounding Data as called by 3D Visualization
    4. Description : Establish connection with the database in order to fetch data at the start.

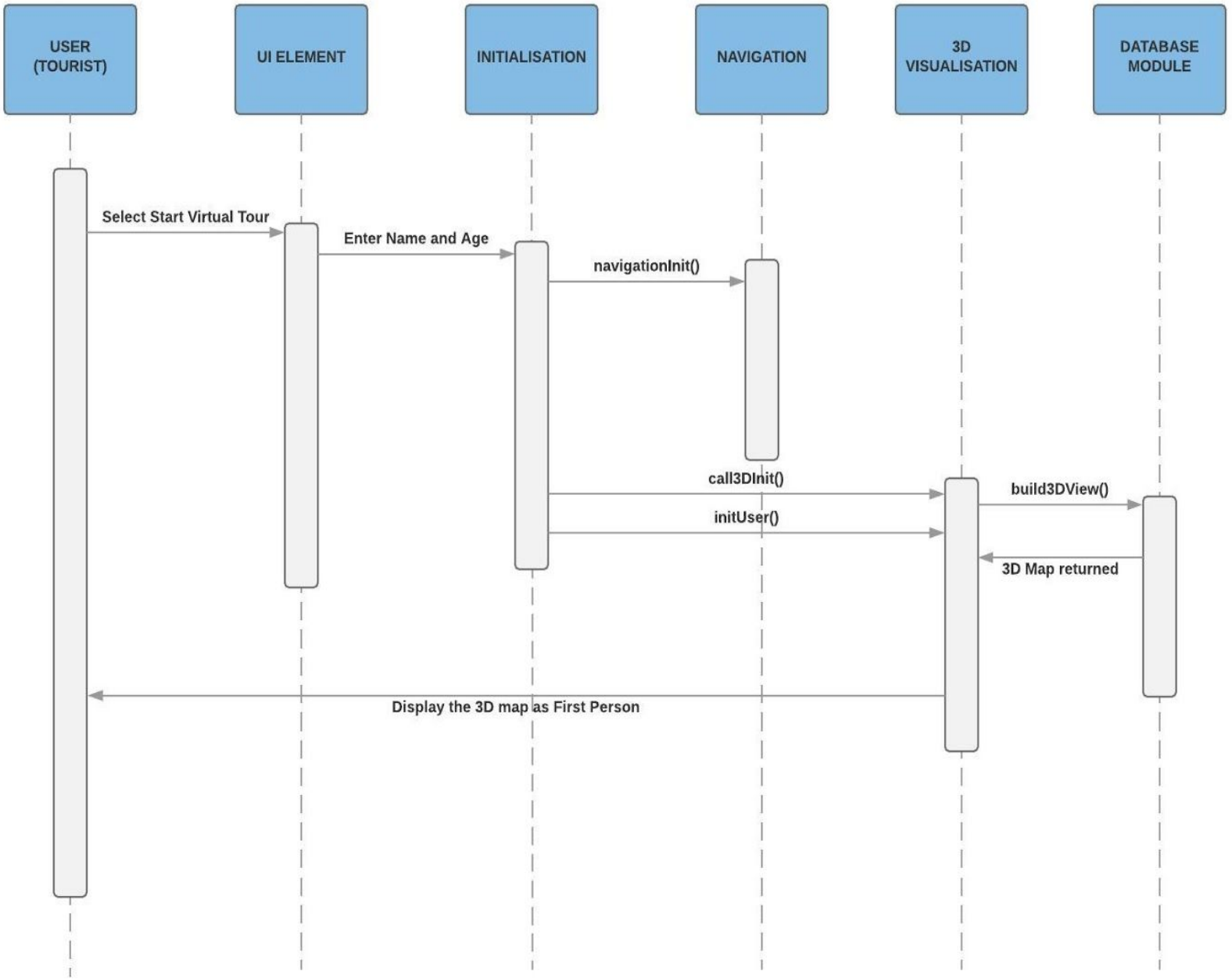
2. get3DObject
    1. Input: Current Coordinates
    2. Called By: 3D Visualization
    3. Output: 3D Object
    4. Description: Returns the 3D Object from Database as and when required by the 3D Visualization whenever there is a change in coordinates.

5 Sequence Diagram

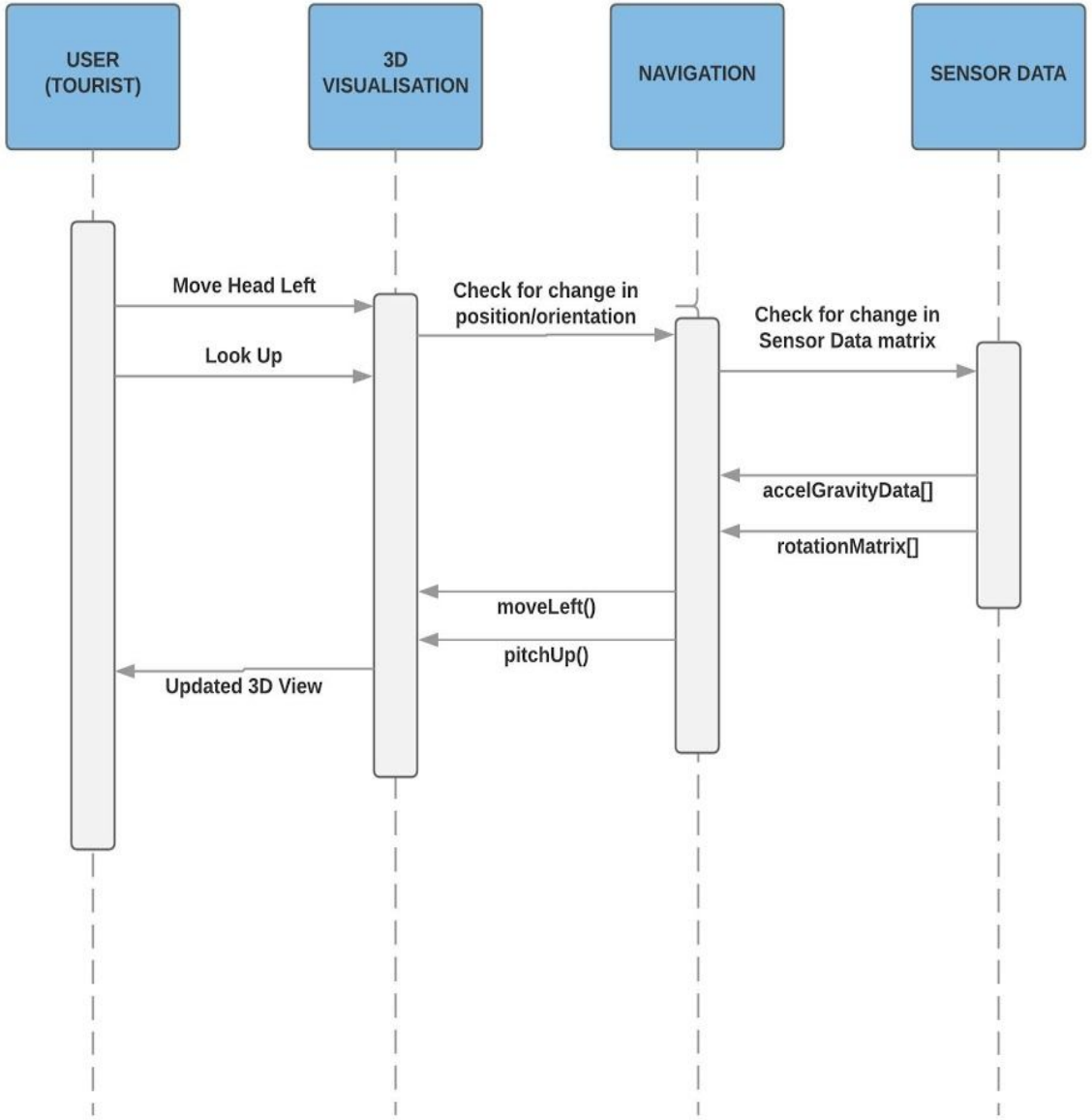
The following Diagrams depict the sequences followed during each of the use cases mentioned.

USE CASE : START VIRTUAL TOUR

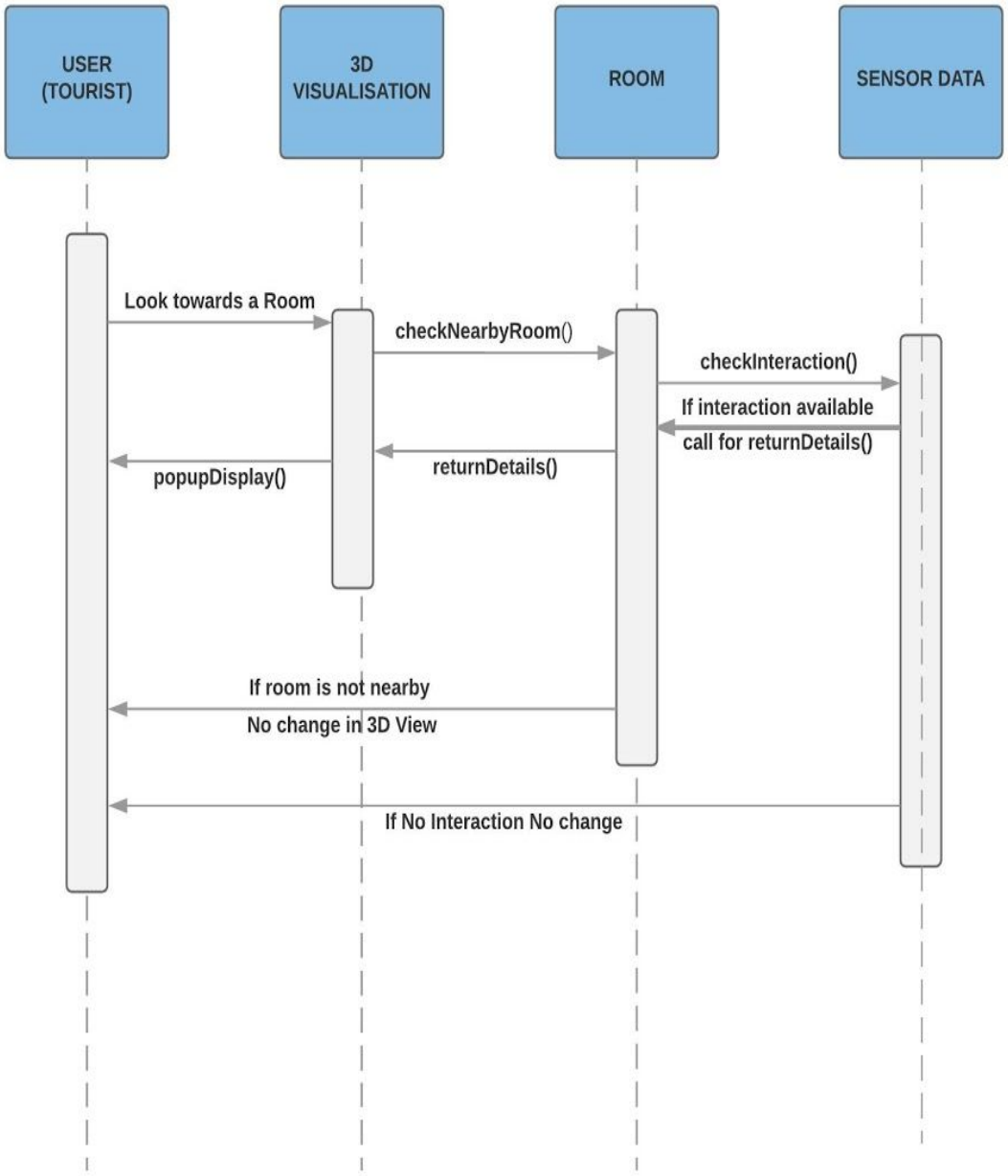
Abhinav Hinger | March 15, 2018



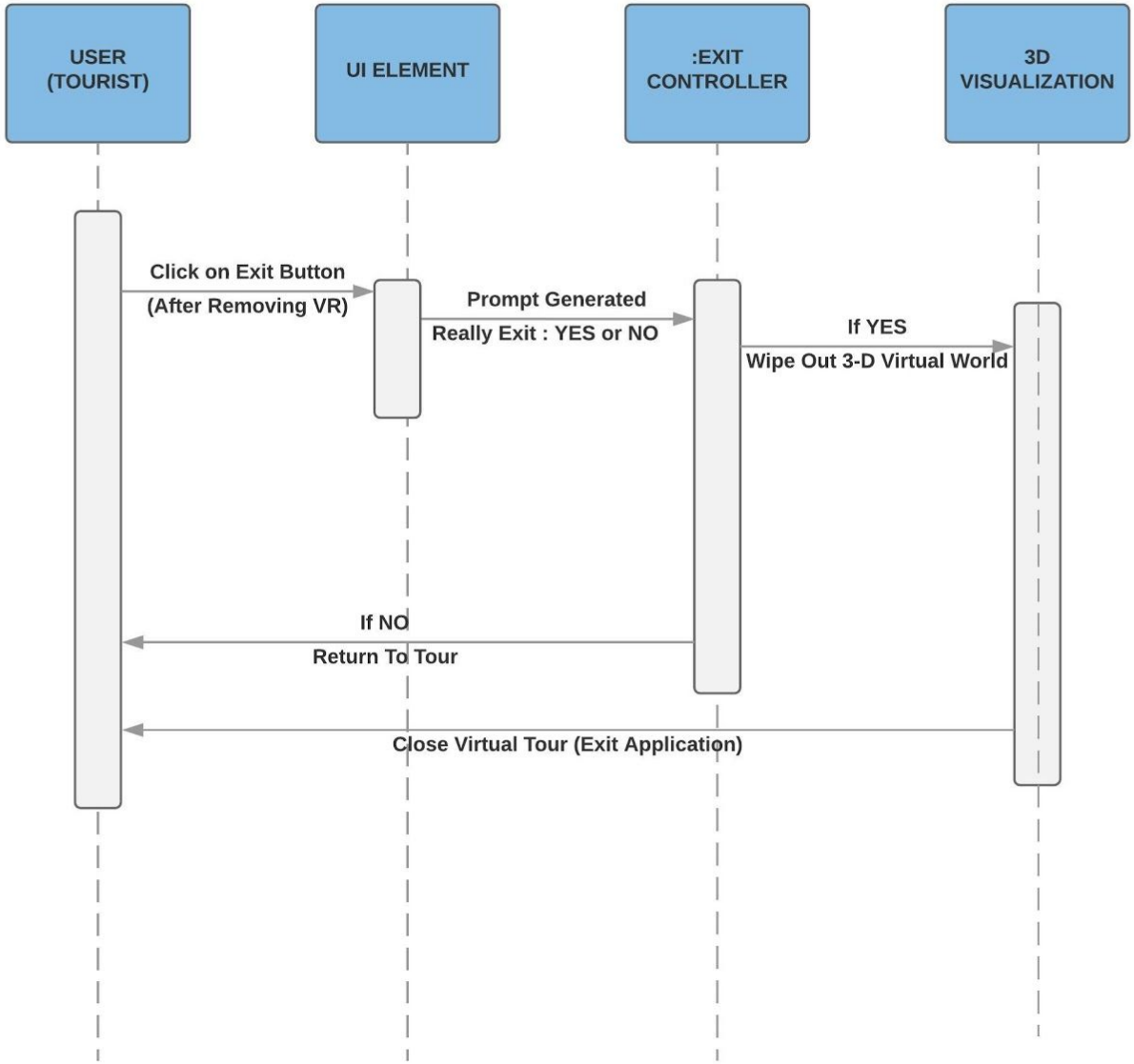
USE CASE : NAVIGATE



USE CASE : VIEW DETAILS



USE CASE : EXIT TOUR

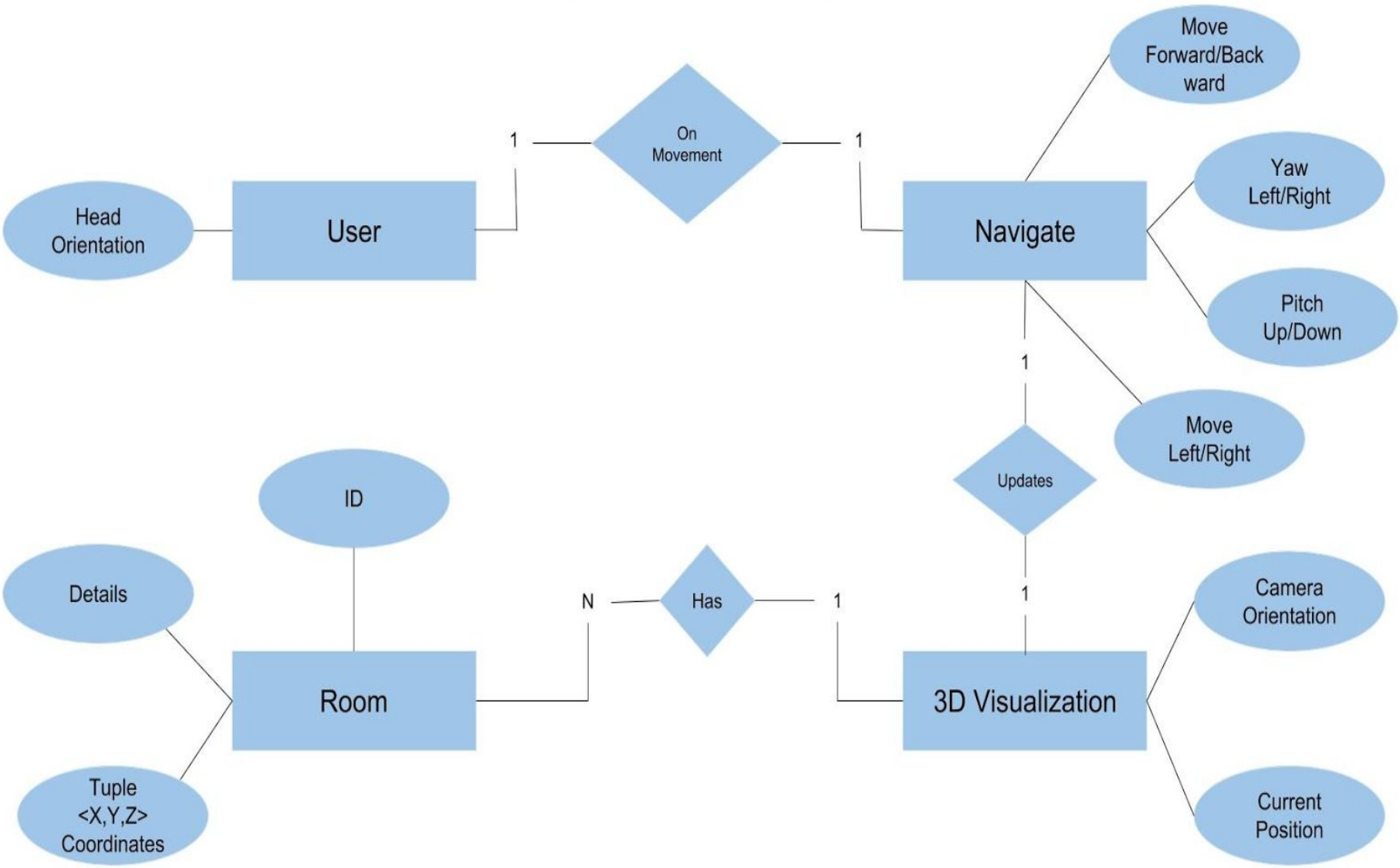


6 Data Design

6.1 Entity Relationship Diagram

ER Diagram is on the next page.

Entity Relationship Diagram



7 References

1. Software Requirement Specification for Virtual Tour of CSE Department dated : 06/02/18
2. Android Developer Reference https://developer.android.com/reference/packages.html
3. Unity Reference https://unity3d.com/learn/tutorials/
4. Lucid Charts https://www.lucidchart.com
5. Blender Reference https://docs.blender.org/manual/en/dev/
6. R. S. Pressman, Software Engineering: A Practioner's Approach, 7th Ed., McGraw Hill, 2010.