Globe Hopper: Tourism in Virtual Reality

Mohammed Ali-Khan
Minah Aqil
Edilberto Carrizales Cavazos
David Favela Corella
Shamsah Hossain
Henry Kim
Jacob Perez

Table of contents

01 Overview

02 Process

03 Requirements

04 Use Case

05 Sequences

06 Class Design

07 Architecture

08 Schedule

09 Cost Estimate

10 Test Plan

11 Similar Designs

12 UI Design

01 Project Overview

Vacation from your living room

Globe Hopper allows users in virtual reality to experience tour experiences uploaded by content creators we partner with

02

Process Model

Incremental Model

- VR Technology is still fairly new
- Allows for user feedback
- Combines structured aspects with agile aspects
- Can expand services over time

03

Project Requirements

Functional Requirements

1. A tourist user shall be able to search for tour experiences by location or landscape.
2. A tourist user will be able to start tour videos, pause them, and exit them.
3. A tourist user will be able to buy tickets for a tour or purchase VIP packages from tour guides.
4. A tourist user will be able to subscribe to a tour guide or agency and will be notified when that guide or agency uploads new content.
5. A tour guide user or agency will be able to upload 360-degree footage and make it searchable by adding a description.

Non-Functional Requirements

1. Usability
2. Performance
3. Space
4. Dependability
5. Security
6. Environmental
7. Operational
8. Development
9. Regulatory
10. Ethical
11. Accounting
12. Safety/security
13. Serviceability

04

Use Case Diagram

Notes:

- Purchase Ticket includes Individual Tour and Tour Package choices
- Login includes account verification stuff
- Book Tour includes search by landscape and location

05

Sequence Diagrams

Login

user
LoginPage
Authentication
Enter User name
Enter Password
Click Login Button
Send user login info
Validate user password
Alternative
[Correct Username and Password]
User Login Accepted
Redirect to target page
Incorrect Username and password
Clear password and inform user that login was denied
User Login rejected

Search for Tour

Upload Tour

Tour Guide Client
uploading site
request to upload
return upload page
select files from disk to upload
Alternative
Sucessful upload
State that upload sucessful
verification and upload
failed to upload
failed to upload, attempt again
Save in folder

Read Reviews

Make Purchase

06

Class Diagram

GglobeHopper
TourList
UserList
ListTours()
UserInformation()
Store()
Support()
1
1..*
Tour
Video
Subtitles
TourGuide
Cost
Location
Reviews
Id
start()
UploadVideo()
AddSubtitles()
UpdateGuide()
UpdateCost()
exit()
pause()
EnableSubtitles()
DisableSubtitles()
Download()
AddReview()
DeleteReview()
ListReviews()
Pause()
ViewAnalytics()
1..*
1
Location
Country
City
Description
Landscape
new()
1
1..*
User
Name
Email
PhoneNumber
UpdateAccountInformation()
Tourist
UserID
Tickets
Packages
Subscribe()
BuyTicket()
PurchaseVIPPackage()
EnableMessages()
DisableMessages()
TourGuide
Tours
EmployeeID
UploadTour()
DeleteTour()

07 Architectural Design

Client-Server Model

Client-Server Model

- Multiple tourist clients connected via virtual reality headset
- Multiple tour guide clients connected via desktop

- Tour Experience Server for holding video
- Catalogue Server for metadata like price and reviews
- Account server for database of user accounts
- Web server for displaying information and organizing the interface

- Client-server is modular
- Designed with scalability in mind

08 Project Scheduling

Schedule

Start Date: January 1st, 2022

There are many who are still hesitant about traveling. Therefore, we would like to offer service soon.

End Date: January 28th, 2022

According to our estimation with the function point algorithm, the first prototype should take about 2 weeks to complete. However, we do not plan on working during weekends and we plan to work 40 hours per week, so our schedule includes extra time to accommodate this as well as any employees needing time off for emergencies.

09

Cost and Effort Estimation

Function Point Estimation

	Function Category	Count	Complexity			Count x Complexity
			Simple	Average	Complex	
1	Number of user input	7	3	4	6	28
2	Number of user output	4	4	5	7	28
3	Number of user queries	4	3	4	6	16
4	Number of data files and relational tables	5	7	10	15	75
5	Number of external interfaces	6	5	7	10	42

GFP: 189

Team Size: 7
Project Duration: 2 weeks

Cost Estimations

Software

- $10,000 annually for Google Earth API
- $14,000 annually

Hardware

- $200-1000 per headset
- $25,000 for computers
- $150 per year for servers

Personnel

- $117,000 per software engineer
- $95,000 per database administrator
- $33,000 per tour guide

10 Test Plan

Unit Test For Login Module

GHLogin.java × GHRating.java

```java
/*
 * Module for Login Interface
 * For Globe Hopper Application
 * Module sets and authenticates user account
 * needs valid username and password
 */
public class GHLogin {

    // private variables for username and password
    private String username;
    private String password;

    // constructor for creating account
    public GHLogin(String u, String p) {
        username = u;
        password = p;
    }

    // central function called to authenticate account
    public boolean startGlobeHopper(String user, String pass) {

        // displays given username and password attempt
        System.out.print("Username: " + user + " " + "Password: ");

        for (int i = 0; i < pass.length(); i++) {
            System.out.print("*");
        }

        System.out.println();

        // if username and password are correct -> login complete
        if (user == this.username && pass == this.password) {
            System.out.println("Login Authenticated\n"
                    + "Welcome to GLOBE HOPPER!\n"); // success
            return true; // completes login
        } else {
            // if username incorrect ...
            if (user != this.username) {
                System.out.println("Login failed - Incorrect username");
                // error message given
            }

            // if password incorrect
            if (pass != this.password) {
                System.out.println("Login failed - Incorrect password");
                // error message given
            }

            System.out.println();
            return false; // login fails
        }
    }
}
```

GlobeHopperLoginTest.java × GlobeHopperRatingTest.java

```java
import static org.junit.Assert.assertEquals;

/*
 * Unit Test for Login Module
 * On Globe Hopper Application
 * since username and password are both sets
 * two possibilities: correct set, or not correct set
 * together there are four possibilities
 * correct username and password
 * correct username with incorrect password
 * incorrect username correct password
 * incorrect username and password
 *
 * both must be correct for success
 */
class GlobeHopperLoginTest {

    String user = "USERNAME";
    String pass = "pass123";

    // new accoutn created with username and password above
    GHLogin gh1 = new GHLogin(user, pass);

    @Test
    void test() {
        // first test checks correct username and password
        assertEquals("Here is the test for login authentication",
                true, gh1.startGlobeHopper("USERNAME", "pass123"));
        // results in success
    }

    @Test
    void test2() {
        // second test checks incorrect username
        assertEquals("Here is the test for login authentication",
                true, gh1.startGlobeHopper("FALSEusername", "pass123"));
        // results in failure
    }

    @Test
    void test3() {
        // third test checks incorrect password
        assertEquals("Here is the test for login authentication",
                true, gh1.startGlobeHopper("USERNAME", "password1234"));
        // results in failure
    }

    @Test
    void test4() {
        // fourth test checks incorrect username and password
        assertEquals("Here is the test for login authentication",
                true, gh1.startGlobeHopper("NOTUSERNAME", "pass456789"));
        // results in failure
    }

}
```

JUnit ×

Finished after 0.225 seconds

Runs: 4/4 Errors: 0 Failures: 3

- GlobeHopperLoginTest [Runner: JUnit 5] (0.005 s)
  - test() (0.001 s)
  - test2() (0.002 s)
  - test3() (0.001 s)
  - test4() (0.001 s)

Failure Trace

```
java.lang.AssertionError: Here is the test for login authentication expected:<true
at GlobeHopperLoginTest.test2(GlobeHopperLoginTest.java:37)
at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
```

Console ×

```
<terminated> GlobeHopperLoginTest [JUnit] C:\Users\mabra\.p2\pool\plugins\org
Username: USERNAME Password: *******
Login Authenticated
Welcome to GLOBE HOPPER!

Username: FALSEusername Password: *******
Login failed - Incorrect username

Username: USERNAME Password: ************
Login failed - Incorrect password

Username: NOTUSERNAME Password: **********
Login failed - Incorrect username
Login failed - Incorrect password
```

Unit Test For Rating Module

GHLogin.java | GHRating.java

```java
/*
 * Module for ratings
 * For Globe Hopper Tour Videos
 * takes input rating from user
 * validates between 0 stars and 5 stars
 * computes average and necessary information
 */
public class GHRating {

    // private variables for rating
    private int count;
    private double AVG;

    // constructor for specific video
    public GHRating() {
        this.count = 0;
        this.AVG = 0;
    }

    // central function for validating rating
    public boolean addRating(int newRating) {

        // if rating is less than 0 stars or greater than 5 stars
        if (newRating < 0 || newRating > 5) {
            System.out.println("Rating must be between 0 and 5 stars\n"
                    + "Cannot be " + newRating); // error message
            return false; // rating addition fails
        }

        // if first entry of video
        if (count == 0) {
            this.count++;
            this.AVG = newRating;
        } else {
            // additional count made
            this.count++;
            // new average computed
            this.AVG = (AVG * (count - 1) + newRating) / (double)count;
        }

        // success message
        // displays new results
        System.out.println("Rating for this Tour is: " + this.AVG + "\n"
                + " Last rating given is: " + newRating + "\n");

        return true; // success
    }
}
```

GlobeHopperLoginTest.java | GlobeHopperRatingTest.java

```java
import static org.junit.Assert.assertEquals;

/*
 * Unit Test for Rating Module
 * On Globe Hopper Tour Video Interface
 * since rating is a range
 * and ratings for this application must be between
 * 0 stars and 5 stars
 * there are 3 possible input values for ratings:
 * values less than 0
 * values between 0 and 5
 * values greater than 5
 * only values between 0 and 5 are valid
 *
 */
class GlobeHopperRatingTest {

    // creates new test video to be rated
    GHRating t1 = new GHRating();

    @Test
    void test4() {
        // first test randomly adds 5 ratings between 0 and 5
        for (int i = 0; i < 5; i++) {
            int r = (int)(Math.random() * 5);
            // since all values between 0 and 5 are accepted ...
            assertEquals("Here is the test for rating authentication",
                    true, t1.addRating(r));
            // results in success
            // and displays the correct information
        }
    }

    @Test
    void test5() {
        // second test adds 5 ratings between -3 and 2
        for (int i = 0; i < 5; i++) {
            int r = (int)((Math.random() * 5) - 3);
            // only values between 0 and 2 will be accepted
            // detection of values less than 0 will terminate loop
            assertEquals("Here is the test for rating authentication",
                    true, t1.addRating(r));
            // results in failure
        }
    }

    @Test
    void test6() {
        // third test adds 5 ratings between 3 and 8
        for (int i = 0; i < 5; i++) {
            // only values between 3 and 5 will be accepted
            int r = (int)((Math.random() * 5) + 3);
            // detection of values greater than 5 will terminate loop
            assertEquals("Here is the test for rating authentication",
                    true, t1.addRating(r));
            // results in failure
        }
    }

}
```

JUnit

Finished after 0.113 seconds

Runs: 3/3 | Errors: 0 | Failures: 2

GlobeHopperRatingTest [Runner: JUnit 5] (0.020 s)
- test4() (0.014 s)
- test5() (0.004 s)
- test6() (0.002 s)

Failure Trace

```
java.lang.AssertionError: Here is the test for rating authentication expected:<true
at GlobeHopperRatingTest.test5(GlobeHopperRatingTest.java:43)
at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
```

Console

```
<terminated> GlobeHopperRatingTest [JUnit] C:\Users\mabra\.p2\pool\plugins\or
Rating for this Tour is: 3.0
 Last rating given is: 3

Rating for this Tour is: 2.5
 Last rating given is: 2

Rating for this Tour is: 1.6666666666666667
 Last rating given is: 0

Rating for this Tour is: 1.5
 Last rating given is: 1

Rating for this Tour is: 1.6
 Last rating given is: 2

Rating must be between 0 and 5 stars
Cannot be -2
Rating for this Tour is: 5.0
 Last rating given is: 5

Rating must be between 0 and 5 stars
Cannot be 6
```

11 Similar Designs

Similar Apps

- Realities
  - 3D touring through computer generated graphics
  - Content created by the app developers
  - Free
  - Supports only Oculus Rift and HTC Vice
- Google Tour Creator
  - 360° photo touring
  - Content created by users
  - Free (discontinued)
  - Support for Google cardboard, Android and desktop.

Similar Apps

- Seven Wonders: Virtual Reality Tours
  - 360° photo tour
  - Fixed content created by the app developers
  - Free
  - Weak VR support
- Sites in VR
  - 360° photo tour
  - Fixed content created by the app developers
  - Free
  - Supports all VR devices

12

User Interface Design

Globe Hopper

Email Address

Password

LOG IN

SIGN UP

Forgot Password?

Globe Hopper

Globe Hopper

Wrapping Up

Thanks!

CS 3354 Software Engineering Group 6

Do you have any questions?

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, and infographics & images by Freepik

Please, keep this slide as attribution