Steven Cole: 000863457

Ekamjot Singh: 000935878

Manminder Singh: 000933795

Roman Sorokin: 000945260

Sukhman Singh Brar: 000929419

Phase 1 – Group 7

**Table of Contents**

[Project Brief 2](#_Toc1059681725)

[Client Information 2](#_Toc1417310849)

[Use Case Diagram 3](#_Toc1817125281)

[Use Case Descriptions 3](#_Toc893059735)

[Class Diagram 7](#_Toc1250830049)

[APPENDIX: Team Constitution 8](#_Toc2011269303)

[The Team 8](#_Toc686151914)

[Team Goals 9](#_Toc674866485)

[IP Statement 10](#_Toc1233303759)

[Member Roles and Specializations 11](#_Toc416809450)

[Task Division 11](#_Toc833002143)

[→ Communication 13](#_Toc104742004)

[→ Meeting Times 13](#_Toc801697430)

[→ Conduct 13](#_Toc1391638283)

[Accountability 14](#_Toc14754053)

[Expelling A Member 17](#_Toc1586484134)

[Conclusion 18](#_Toc1344977660)

[Agreement 18](#_Toc394041164)

# **Project Brief**

The Destination History App focuses on the tourism business, providing tourists with information on destinations and points of interest in the secondary tourism market: small communities. This is the market with the greatest potential for growth in Alberta. The plan is to start with Alberta and move on to the rest of Canada once the app and the business are stable.

The application will be on a mobile interface. Some **must-have features** would include user/client authentication, a subscription service ($9/month after a 3-month free trial), payment integration via credit card or PayPal, a location-based search using Google Maps, a business listing display, push notification, a proper database management system, and multilingual support (English, French, and Spanish). The app would include a separate GUI layout for clients (with text editors and templates) and users (with image displays, search functionality, and business information)

**Nice-to-have features** would include the ability to customize a web page, SEO optimization, advertising inserts, enhanced UI/UX elements, administrative features, and social media integration.

The system will include a tech stack of JavaScript, React/React Native, Node.js, SQL, and APIs like Google Maps. Prototype design will be completed via Figma.

# **Client Information**

1. Contact Information
   1. Collings, Mark (Organization Head)
   2. Phone: 403-598-3301
   3. Email: [mark.collings@destinationhistory.ca](mailto:mark.collings@destinationhistory.ca)
2. Location
   1. Red Deer, Alberta
3. Type of Organization
   1. Tourism Industry

# **Use Case Diagram**

A diagram of a diagram

AI-generated content may be incorrect.

# **Use Case Descriptions**

|  |  |
| --- | --- |
| **Use Case** | View and Search Content |
| **Actor** | User/Tourist and Business Client |
| **Description** | Users and Business Client can explore a variety of historical sites |
| **Preconditions** | * The user and the client must have access to the application * The user must have a stable connection |
| **Postcondition** | The search history will be saved for further use in future |

|  |  |
| --- | --- |
| **Use Case** | Manage Account |
| **Actor** | User/Tourist and Business Client |
| **Description** | User and Business Client can manage their personal account |
| **Preconditions** | * The user and the client must be logged in. * The account must exist. |
| **Postcondition** | The information can be saved in the database or the system. |

|  |  |
| --- | --- |
| **Use Case** | Create Account |
| **Actor** | User/Tourist and Business Client |
| **Description** | User and Business Client can register for their new accounts |
| **Preconditions** | * The user and the business client must provide valid details. * The email or the number must be unique. |
| **Postcondition** | The profile is created in the system. |

|  |  |
| --- | --- |
| **Use Case** | Manage alert |
| **Actor** | User/Tourist |
| **Description** | User can set or get alerts for the destinations |
| **Preconditions** | The user must be logged in. |
| **Postcondition** | The system should update notifications according to the user preference |

|  |  |
| --- | --- |
| **Use Case** | Login |
| **Actor** | User/Tourist and Business Client |
| **Description** | Users and Business Client can access their account by logging in. |
| **Preconditions** | Account Registration is required. |
| **Basic flow** | The customer needs to fill in these details for Login in the system: -   * Login name * Password   (Login name is also considered as unique id for customer) |
| **Postcondition** | Users will be able to browse locations. |

|  |  |
| --- | --- |
| **Use Case** | View/Search on map |
| **Actor** | User/Tourist |
| **Description** | Users can use the map to view locations and navigate more effectively. |

|  |  |
| --- | --- |
| **Use Case** | View Business info and Images |
| **Actor** | User/Tourist |
| **Description** | View will discover about the business, including photos. |
| **Preconditions** | The user must have the access to view the business details |
| **Postconditions** | The user can successfully view the details |

|  |  |
| --- | --- |
| **Use Case** | Save to favourites |
| **Actor** | User/Tourist |
| **Description** | User can bookmark business or destinations they are interested in. |

|  |  |
| --- | --- |
| **Use Case** | Change Language |
| **Actor** | User/Tourist |
| **Description** | Users have the option to select a preferred language for easier access. |
| **Preconditions** | The app must support multiple languages |
| **Postconditions** | App display the content in the selected language |

|  |  |
| --- | --- |
| **Use Case** | Social Media Integration |
| **Actor** | User/Tourist and Business Client |
| **Description** | The user and Business Client can use this option to link the destination or to share their content through social media platforms |
| **Preconditions** | The system must support social media connectivity. |
| **Postconditions** | The user and the business client are able to share the content on social media |

|  |  |
| --- | --- |
| **Use Case** | Manage/Update Business info |
| **Actor** | User/Tourist |
| **Description** | Business clients can update and control their information within the app. |
| **Preconditions** | The user and the business client must be logged in. |
| **Postconditions** | The info will be updated successfully. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case** | View Analytics | | |
| **Actor** | Business Client and Administrator | | |
| **Description** | Clients and Administrators can monitor traffic and what Users look at | | |
| **Precondition** | Must be logged in | | |
| **ACTION** | | **SYSTEM RESPONSE** |
| * 1 Select:   ‘*View Analytics*’ button | | * 2 Displays data (to be discussed with (Mark and Team to determine relevant data) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case** | Manage Subscription | | |
| **Actor** | Business Client | | |
| **Description** | Clients can manage their subscription, such as: 1-month, auto-renew, cancel | | |
| **Preconditions** | Client must be logged in to access subscription information | | |
| **Postcondition** | Subscription type changed based on selection | | |
| **ACTION** | | **SYSTEM RESPONSE** |
| * 1 Select:   ‘*Subscription*’ button | | * 2 Displays subscription status screen and the option to Update status |
| * 3.1.1 Select: ‘*Update*’ | | * 3.1.2 Prompt to select either: > Make next payment (3.1.3) > Automatic Renewal (3.1.4) > Turn off Automatic Renewal (3.1.5) |
| * 3.1.3 Select: ‘*Make next payment*’ | | * 3.1.3-2 Proceed to payment \* |
| * 3.1.4 Select: ‘*Automatic Renewal*’ | | * 3.1.4-2 Turns on Automatic Renewal |
| * 3.1.5 Select: ‘*Turn off Automatic Payment*’ | | * 3.1.5-2 Turns off Automatic Renewal |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case** | Process Payment \* | | |
| **Actor** | Business Client | | |
| **Description** | Client makes a payment to the system to continue their subscription | | |
| **Preconditions** | Client must have their payment method set up and account verified | | |
| **Postcondition** | Payment processed and subscription maintained | | |
| **Alternative** | Payment failed – warning sent to client followed by their subscription ending | | |
| **ACTION** | | **SYSTEM RESPONSE** |
| * 1 Payment Selected | | * 2 Process Payment:  > On Success: 2.1 > On Failed: 2.2 |
|  | | * 2.1: “Transaction Complete” message * Complete transaction and return to main page |
|  | | * 2.2: “Transaction Failed” message * Incomplete transaction and prompt to try again soon, then return to main page |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case** | Manage Events | | |
| **Actor** | Business Client | | |
| **Description** | Clients can add/update the location to highlight events/activities taking place | | |
| **Precondition** | Client is logged in and has an active subscription | | |
| **Postcondition** | Event/Activity information updated | | |
| **ACTION** | | **SYSTEM RESPONSE** |
| * 1 Select ‘*Add Event*’ | | * 2 Prompt Client to input event information * Asks client to confirm information > On confirm: 3.1 > On edit: 3.2 > On cancel: 3.3 |
| * 3.1 Confirm | | * 3.1-2 Event submitted and posted to the location |
| * 3.2 Edit | | * 3.2-2 Return to the event form to modify event information (2) |
| * 3.3 Cancel | | * 3.3-2 Discards form and returns to main page |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case** | Manage Users and Clients | | |
| **Actor** | Administrator | | |
| **Description** | Admin can keep track of and make changes to all Users and Clients | | |
| **Precondition** | Admin is logged in | | |
| **ACTION** | | **SYSTEM RESPONSE** |
| * 1 Selects a User or Client | | * 2 Displays User or Client account or post details |
| * 3.1 Select: ‘*Modify*’ | | * 3.1-2 Change account or post details > Save / Cancel |
| * 3.2 Select: ‘*Delete*’ | | * 3.2-2 Deletes account or post access/information (Account information is kept in the database) |
| * 3.3 Select: ‘*Return*’ | | * 3.3-2 Returns to the previous page |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case** | Manage Systems | | |
| **Actor** | Administrator | | |
| **Description** | Admin makes sure the system is working through updates to the application, monitoring metrics, handle backups and recovery, etc. | | |
| **ACTION** | | **SYSTEM RESPONSE** |
| * 1 | | * 2 |

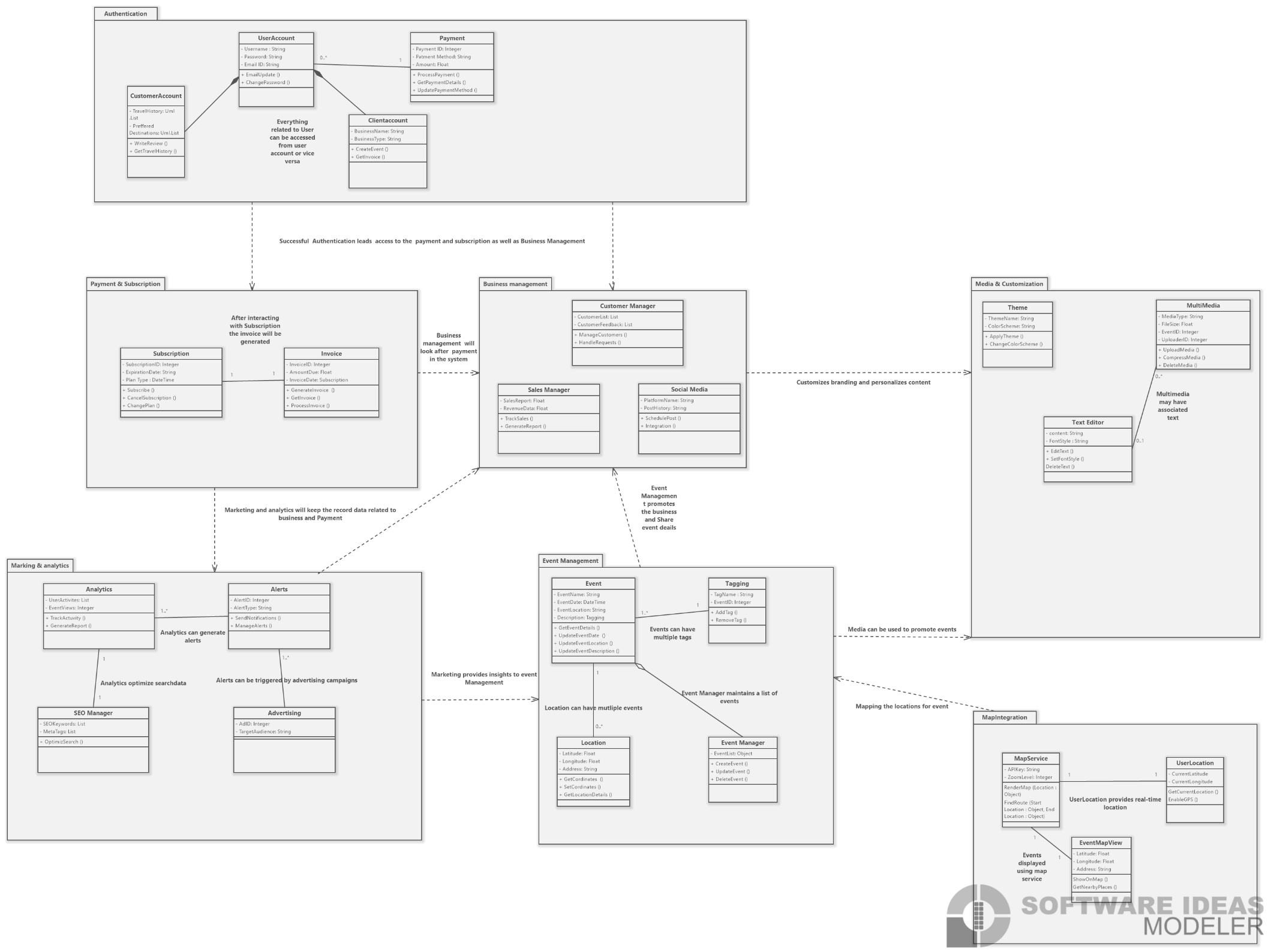
|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case** | Moderate Content | | |
| **Actor** | Administrator | | |
| **Description** | Admin ensures content remains consistent throughout the platform and has the choice to approve or deny requests to post on the app | | |
| **ACTION** | | **SYSTEM RESPONSE** |
| * 1 Selects a User or Client | | * 2 Displays User or Client account or post details |
| * 3.1 Select: ‘*Modify*’ | | * 3.1-2 Change account or post details > Save / Cancel |
| * 3.2 Select: ‘*Delete*’ | | * 3.2-2 Deletes account or post access/information (Account information is kept in the database) |
| * 3.3 Select: ‘*Return*’ | | * 3.3-2 Returns to the previous page |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case** | Verify Users and Clients | | |
| **Actor** | Administrator | | |
| **Description** | Admins verify account information related to Users and Clients before they can have full access to the application | | |
| **Precondition** | Admin is logged in & received account information from a Client | | |
| **Postcondition** | Client information is verified | | |
| **Alternative** | Client information is rejected – Client must ask for verification again | | |
| **ACTION** | | **SYSTEM RESPONSE** |
| * 2 Compares information found online (maybe contacts Client) | | * 1 Message received from Client with account data |
| * 3 Confirms data and submits verification | | * 4 Client verification > On Success: 5.1 > On Failure: 5.2 |
|  | | * 5.1 Success: Message sent to Client and account is verified |
|  | | * 5.2 Failure: Message sent to Client stating why the verification failed |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case** | Update System | | |
| **Actor** | Administrator | | |
| **Description** | Update various areas of the system, such as: security, information, software, etc. | | |
| **ACTION** | | **SYSTEM RESPONSE** |
| * 1 | | * 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case** | SEO Optimization | | |
| **Actor** | Administrator | | |
| **Description** | Administration can optimize business titles and descriptions and add relevant keywords so that they become easier to search on the application | | |
| **ACTION** | | **SYSTEM RESPONSE** |
| * 1 | | * 2 |

# **Class Diagram**

****

**APPENDIX: Team Constitution**

**Tour.js**

A yellow and white logo

AI-generated content may be incorrect.

**The Team**

**Manminder Singh**

Manminder is a software development student that has interest in both the frontend and backend. He has worked in multiple projects in college and made some personal ones as well. During this time, he has learned some programming languages and is trying to master them.

**Roman Sorokin**

Roman has a strong foundation in data analytics and backend development, with strengths in database management systems and data analysis, allowing him to excel in languages like SQL and Python and systems like Oracle. At his current employer, one of his roles is analyzing organizational data for resource allocation and securing continued funding through data-driven decision making.

**Ekamjot Singh**

Ekamjot is focusing on full-stack development, having strong foundations in JavaScript, React, Oracle, MongoDB with Node.js. He is working to improve my skills in DSA using Java and in Mobile Development using React Native.

**Steven Cole**

Steven is a Software Development student with a focus on User Experience, Design, and Project Management. He has committed himself to developing strong time-management and communication skills. Outside of student life, his time is spent with his wife and daughter, two jobs, and designing games.

**Sukhman Singh Brar**

Sukhman has a strong analytical mindset, excellent problem-solving abilities, and a passion for learning new technologies. He began to learn new programming languages and has a strong touch in mobile and web development using React.

**Team Goals**

Grade we hope to achieve: A

Skills/accomplishments we aim to gain through the process:

1. Gain experience in building a cohesive full-stack mobile application from start to finish
2. Building collaborative skills and improving our ability to work as a team in various roles
3. Developing skills to work and communicate with real-world clients
4. Improve skills in React/React Native
5. Build portfolio-worthy work that can be demonstrated in the future with potential employers
6. Gain a practical understanding of the Software Development Lifecycle
7. Improve the economies of lower-profile tourist destinations/locations

**IP Statement**

The Destination History App and all its associated materials are owned by Destination History Inc. (the client) including the application’s source code and architecture, business information, database designs and implementations, UI/UX design, features and functionalities, brand assets and marketing materials.

Members of Tour.js will develop the code and materials for the project which, upon completion, would be transferred to Destination History Inc. The members must document their contributions to ensure the scalability of the project. The members can include this project in their portfolio but will be unable to reuse the code for commercial purposes.

Destination History Inc. will have exclusive rights to commercialize the application upon completion, modify and update the software as it sees fit, create derivatives of the project, license the technology, and handle all future business arrangements.

Client confidentiality and all project’s details will be treated as confidential during developments. Tour.js members, if needed, will sign appropriate non-disclosure agreements.

Sait Policy Links:

* **AC.2.11** [Intellectual Property Policy](https://www.sait.ca/assets/documents/about-sait/policies-and-procedures/academic-student/ac-2-11-intellectual-property.pdf)
  + **AC.2.11.1** [Intellectual Property Procedure](https://www.sait.ca/assets/documents/about-sait/policies-and-procedures/academic-student/ac-2-11-1-intellectual-property.pdf)
    - **AC.2.11.1** [Schedule A - Technology Disclosure Form](https://www.sait.ca/assets/documents/about-sait/policies-and-procedures/academic-student/ac-2-11-1-schedule-a.pdf)
  + **AC.2.11.2** [Revenue Sharing and Commercialization Procedure](https://www.sait.ca/assets/documents/about-sait/policies-and-procedures/academic-student/ac-2-11-2-revenue-sharing-and-commercialization.pdf)

**Member Roles and Specializations**

**Manminder Singh**

|  |  |  |
| --- | --- | --- |
| Full Stack Specialist | Secondary Mediator | FRONTEND (Tailwind CSS, react native)  BACKEND (JavaScript, react) |

**Roman Sorokin**

|  |  |  |
| --- | --- | --- |
| Database Developer | Database Design/Management | Backend Development (Python) |

**Ekamjot Singh**

|  |  |  |
| --- | --- | --- |
| Full Stack Specialist | FRONTEND (Tailwind, React Native)  Backend (Node.js) | No-SQL Database (MongoDB)  SQL (Oracle) |

**Steven Cole**

|  |  |  |
| --- | --- | --- |
| Project Manager | Mediation Lead | User Interface/Experience Designer |

**Sukhman Singh Brar**

|  |  |  |
| --- | --- | --- |
| Full Stack Specialist | FRONTEND (Tailwind, React Native) | BACKEND (JavaScript, react) |

**Task Division**

All roles can be interchanged/swapped, and personnel can be added or removed if necessary.

Project Management

*Ekamjot* and *Steven* will oversee the project by taking on being the main point of contact with the client, ensure deadlines are met, and team members communicate properly and effectively.

Main roles would include:

* Creating project timelines to ensure successful completion of each step of the software development lifecycle
* Scheduling weekly/bi-weekly team meetings and check-ins either on Teams or Discord
* Communicating and setting up meetings with the client via Teams or in-person
* Ensuring the project remains within the scope of the class and relevant features are prioritized (i.e., must-have features).

Deliverables/milestones would include:

* The completion of a project timeline document via MS Project
* Creating meeting schedules and agendas for team meetings
* Writing meeting minutes and action items during Teams meetings with the client

UI/UX Design

*Steven* can oversee designing the app’s user interface and optimizing user experience. This can be done through tools like Figma or other relevant tools.

Main roles would include:

* Working on wireframes and prototypes for the app that can be demonstrated to the team and to the client.
* Design visually attractive and easy-to-use layouts
* Ensure the app is easy to access and keeps accessibility in mind

Deliverables/milestones would include:

* Creating a wireframe protype for all app screens (user/client interfaces) using Figma
* Defining the design of the application (colors, font, etc.)

Frontend Development

*Manminder* and *Sukhman* will oversee developing the client-side of the app.

Main roles would include:

* Implementing the UI/UX designs into functional screens.
* Ensure smooth interactivity between features (e.g., maps, looking at business info, etc.).

Deliverables/milestones would include:

* Creating functional user interface components
* Ensure the client-side works properly on all devices (e.g., iOS, Android).
* Writing a frontend codebase with proper documentation

Backend Development

*Roman* and *Ekamjot* will oversee the development of the server-side logic, databases, and APIs.

Main roles would include:

* Choosing and setting up a database management system to store user and client data, attractions, analysis (page views, visits), payment information, etc.
* Developing and integrating APIs for communication between the frontend and backend.
* Implementing user authentication and security features.
* Integrating third-party services (e.g., Google Maps API, payment gateways, etc.)

Deliverables/milestones would include:

* Creating a database schema and architecture through entity relationship diagrams
* Writing API documentation with endpoints
* Setting up an authentication system for user, client, and administrator accounts
* Creating a backend codebase with proper documentation

Documentation and Reporting

*Steven* will oversee documenting the project process and creating user guides for app users.

Main roles would include:

* Writing technical documentation for future developers.
* Creating user manuals or FAQs for app users.
* Prepare progress reports for stakeholders.

Deliverables/milestones would include:

* Ensuring that all code has proper inline comments and external documentation explaining the code structure
* Figuring out common issues with the application and using it to write an FAQ or troubleshooting document.
* Show summaries of completed work regularly at meetings.

**Team Expectations**

**→ Communication**

**Within Team**

Main mode of communication: Teams

Secondary mode of communication: Discord

Required time to reply:

* Before next meeting (unless marked ‘urgent’ or approaching deadlines).

**\*Urgent must be within 24-48 hours**

* If unable to reply in designated time, the question should be reiterated in the meeting, and an explanation should be provided as to why the member was not able to reply.
  + If the member who failed to reply also misses the meeting, this will count as an offence but may be retracted if the group deems that the eventual provided reason is justified.

**With Client**

We will primarily communicate with the client by email for updates, questions, and any document sharing. Microsoft Teams will be our main application for virtual meetings. Replies by email should be completed within 24-48 business hours upon receipt of the client’s email.

In-person meetings will be held, when necessary, in Calgary at SAIT or wherever convenient for the client. We will schedule regular meetings with the client every 2-3 weeks, with the frequency of meetings depending on the number of issues/questions that need to be discussed with the client. A minimum 48 hours' notice needs to be provided for scheduling meetings.

**→ Meeting Times**

Meeting time #1: Touch base in-person on Wednesday, after class from 7:50-8:00pm.

Meeting Time #2: Online on Friday after class from 7:50-8:00pm.

**\*Meetings subject to change as needed and change as deadlines approach**

**→ Conduct**

Key Points of Conduct

|  |  |
| --- | --- |
| * **Punctuality** | On time or let the team know that you will be running late. |
| * **Attendance** | In class/online and at required meetings. If absent, let the team know as soon as possible. |
| * **Communication** | Responding to group and direct messages via Teams and Discord within 24 hours. (Even if it is just using a reaction). |
| * **Participation** | Performing tasks and role responsibilities and fulfilling their responsibilities as a member of Tour.js. |

**\*If members are having difficulty communicating with the group – reach out to the instructor via Teams**

**Accountability**

**Punctuality*****/ Attendance***  
 *Tier 1 Offence*   
 Explanation must be provided to Steven and/or Manminder as to what caused this level of offence. (either in person or via Teams/Discord).

*Tier 2 Offence*

The Offender will be spoken to regarding the offence.

*Tier 3 Offence*

Vote will be called -> On a failed vote: This will be treated as a Tier 2 Offence. successful vote: The Offender must provide a suitable means of making amends as approved by Steven and/or Manminder

*Causes for Penalty: Consistently late, Missing meetings or class (without providing a reason), or Related Issue*

**Communication**  
 *Tier 1 Offence*

Explanation must be provided to Steven and/or Manminder as to what caused this level of offence. (either in person or via Teams/Discord).

*Tier 2 Offence*

The Offender will be spoken to regarding the offence.

*Tier 3 Offence*

Vote will be called -> On a failed vote: This will be treated as a Tier 2 Offence. On a successful vote: The Offender must provide a suitable means of making amends as approved by Steven and/or Manminder.

*Causes for Penalty: Not communicating with team or in discussions, not posting a reaction (in Teams or Discord) to indicate that you have read/acknowledged the message*

**Participation**  
 *Tier 1 Offence*

The Offender will be spoken to regarding the offence.

*Tier 2 Offence*

Vote will be called -> On a failed vote: This will be treated as a Tier 1 Offence. On a successful vote: The Offender must provide a suitable means of making amends as approved by Steven and/or Manminder.

*Tier 3 Offence*

Vote will be called -> On a failed vote: This will be treated as a Tier 2 Offence. On a successful vote: The professor will be notified, and that student will have 25% taken off from their final project grade. If the group member continues to break the code of conduct, additional marks will be docked.

*Causes for Penalty: Not completing assigned works, not contributing to individual or group related things/activities*

**Example Conflicts**

*Case 1*

Sukhman and Steven get into an intense argument on whether Sukhman getting sick 3 times in a row, resulting in a month's worth of non-contribution should dock him marks from the assignments within the period.

1. Since Steven is typically main mediator, the second appointed mediator (Manminder), will talk to each of the involved parties and ask for their stories and arguments.
2. Gather the group members (excluding involved parties) and discuss what the majority believes is the proper action.
3. Based on our democracy, a decision will be taken on how to respond to Sukhman’s lack of participation.
4. Both Sukhman and Steven must accept the decision of the group.

*Case 2*

Roman is met with one of his close relatives in urgent medical care. He explains to the group of his sudden family emergency but insists that he will be able to complete his share of the work despite the situation. The group reluctantly agrees. As a few weeks pass and the deadline of a heavily weighted assignment draws near, Roman has not completed an iota of work, leaving the assignment with a large hole that is detrimental and will lead to a failing mark.

1. Express empathy and understanding of family situation to Roman to maintain supportive and open communication.
2. Gather the group members to discuss the impact of his absence to the assignments.
3. If it becomes evident that Roman is unable to contribute due to ongoing challenges, reevaluate the initial agreement of him sticking to his part. Alternatives may include redistribution of Roman’s tasks to group members.
4. Reach out to instructor explaining the situation in hopes for an adjustment to deadline, or an extension.
5. If Roman is still unable to contribute, make sure to document efforts made to include him.
6. Complete assignment and inform instructor.

*Case 3*

3 days before an important deadline, Ekamjot is taking a stroll in a certain park when he is suddenly kidnapped and trafficked into Belarus. His sudden disappearance makes him unable to reply to his groupmate’s messages, and most of the assignment is stored on his device. This results in the group with no work done, and a kidnapped teammate.

1. Try to communicate and reach out to Ekamjot through any means.
2. Document his sudden disappearance and attempts of contact.
3. Try and divide the work and attempt to finish assignment.
4. Inform instructor about the situation after making efforts and plan to complete assignment without Ekamjot.
5. Ask for deadline adjustments.
6. A group vote and discussion will be held for Ekamjot’s consequences.

**Expelling A Member**

If the team is concerned about a group member missing deadlines, providing poor quality work, missing class/team meetings, or failing to communicate the first attempt would be to try to resolve this issue within the team given the means discussed earlier. If no resolution is met, a final warning will be given, and the instructor would be notified. In extreme situations (e.g., 5+ days without communication, hostile behavior, academic dishonesty, etc.) the instructor will be contacted to intervene immediately, potentially leading to the expulsion of the group member.

**Conclusion**

Our team consists of 5 members with diverse skills in full-stack, database, and project management that will make it easier to complete this project well and on time. There is a clear IP agreement set with Destination History Inc., detailed communication and accountability systems, and well-defined conflict resolution procedures.

**Predicted Successes**

* Satisfaction of both the client and the team upon completion of a full-stack mobile application thanks to a wide skill diversity of team members and establishment of frequency and mode of communication
* Timely progress due to an existence of a clear accountability system
* Proper response times within the team with the help of multiple communication channels
* Proper conflict resolution due to well-defined procedures

**Potential Challenges/Failures**

* Overlapping of roles due to some share in responsibility/division of tasks
* Balancing client and academic requirements
* The ambitions of the project compared to team member expertise

**Overall Assessment**

With proper procedures, policies, and task division in place, we believe that we have a strong foundation for success and demonstrate maturity in process development. Looking forward, we are in a good place to overcome challenges if we keep an open mind and focus on the ultimate goals of the project.

**Agreement**

We all agree to follow this team contract through the CPSY-301 course and are all responsible for enforcing the rules outlined above.

|  |  |
| --- | --- |
| **Roman Sorokin** | February 15, 2025 |
| **Manminder Singh** | February 15, 2025 |
| **Sukhman Singh** | February 15, 2025 |
| **Ekamjot Singh** | February 15, 2025 |
| **Steven Cole** | February 15, 2025 |