



**UNIVERSITY  
OF LONDON**

**Student Number:** 200342355

**Name and Surname:** Susanna Ventafridda

**Degree Title:**  
BSc Computer Science

**Course/Module Title:** Computer Science Final Project

**Course/Module Code:** CM3070

**Date:** September 19<sup>th</sup> 2023

**Project Title:** GroupTripper – A web application to plan trips collaboratively

## Table of Contents



**UNIVERSITY  
OF LONDON**

.....	0
Introduction.....	4
Template chosen .....	4
Motivation for the project.....	4
Domain and users.....	4
Literature review – Market research.....	6
SWOT analysis .....	7
Design .....	8
Requirements and acceptance criteria.....	8
Use case diagrams .....	8
Prototypes .....	8
Technical infrastructure .....	8
Back-end .....	8
Front-end.....	10
API design .....	10
Data store design.....	10
Realtime web technologies .....	10
Monetization .....	11
Legal considerations.....	11
Search Engine Optimization .....	11
Evaluation plan .....	11
Development plan & timelines.....	11
Implementation.....	12
Entry point and Routers .....	12
Ajax calls .....	12
Hosting and Firestore database set-up .....	12
Authentication.....	12
Bootstrap 5 .....	13
Input validation and sanitization.....	14
Database interactions and security.....	14

Realtime technologies .....	14
Useful libraries used .....	14
Evaluation .....	15
Whitebox testing - Unit tests.....	15
Blackbox testing.....	16
Vulnerability checks.....	16
User testing.....	16
Planning vs. execution evaluation.....	17
Conclusion .....	17
References .....	19
Appendix.....	26
Appendix A – Market research.....	26
A.1 – Competitors analysis summary .....	26
A.2 – Visual collection of competitors' solutions .....	32
Appendix B – Functional requirements and acceptance criteria .....	63
Appendix C – Medium-fidelity prototypes .....	72
C.1 - First iteration .....	72
C.2 - Second iteration .....	76
C.3 - Third iteration.....	85
Appendix D – Email sending costs .....	90
Appendix E – Addresses autocomplete costs.....	91
Appendix F – Firestore collections design .....	92
F.1 – Users .....	92
F.2 - Trips .....	92
Appendix G – Payment platforms comparison.....	94
Appendix H – Features available with Pro subscription .....	94
Appendix I – UML Diagrams .....	95
Appendix J – Technical infrastructure .....	96
Appendix K – REST API.....	96
Appendix L – Timelines and Task Breakdown .....	97
L.1 – Gantt Chart.....	97
L.2 – Critical Path Analysis .....	99
L.3 Updated Gantt chart.....	99
Appendix M – SEO techniques .....	100

Appendix N – Feature prototype unit tests.....	102
N.1 package.json configurations .....	102
N.2 – unit tests folders .....	103
N.3 – Example unit tests for routes.....	104
N.4 Unit tests results .....	105
Appendix O – Feature prototype database interactions.....	106
O.1 – DB class: contains generic methods to interact with Firestore .....	106
O.2 – Sample queries.....	107
Appendix P – Feature prototype input validation and sanitization .....	108
P.1 - Front-end validation.....	108
P.2 – Backend sanitization.....	109
Appendix Q – Authentication .....	110
Q.1 – session creation API end point.....	110
Q.2 – Frontend Login call.....	111
Appendix R – GitHub repository and Git usage .....	112
R.1 - git log.....	112
R.2 – GitHub repository .....	113
Appendix S – Blackbox testing.....	113
Appendix T – Security scan results .....	121
T.1 results view – first scan.....	121
T.2 OpenVAS– first scan.....	122
T.3 Nmap – first scan .....	123
T.4 OWASP– first scan .....	124
T.5 – second scan.....	125

## Introduction

As a final project, I am going to build a collaboration web application that allows to plan and track trips as a group more effectively: GroupTripper.

With this project I want to demonstrate my ability to build a modern and secure web application that leverages cloud technology, integrates with third party APIs, optimizes SEO, and considers monetization strategies including hosting advertisements and allowing in-app payments via Stripe. By the end of this project, I want to have developed the POC of a product that could be launched successfully on the market and be a potential source of income. I am however aware of legal risks I could be subjected to by launching a business as a private individual and will thus take measures to ensure the product usage is restricted for the time being only to individuals who possess a private code.

### Template chosen

The template I have chosen is project Idea Title 2: A Collaboration web application, from CM3035 Advanced Web Development.

### Motivation for the project

Recent statistics about travel support the need for this application. According to TravelPerk [1], most American adults prefer booking their trips online, especially on their phones. Additionally, TravelPerk reports that travelers visit an average of 38 websites to complete and book their travel plans. Furthermore, TravelPerk highlights that the biggest frustration standing in the way of planning a trip is the need to compare different options.

The market has only recently started to address this need of adults to plan their own trips in a systematic and simpler way [4, 6] and hence there is an opportunity to enter this niche by offering a product that strategically bridges the gaps present in the existing solutions offered by the competition. I describe the details of my value proposition in the “Literature review – Market research” section of this report, which includes a SWOT analysis.

### Domain and users

GroupTripper aims to be the best social network application on the market that facilitates planning trips as a group online, minimizing the so called “analysis paralysis” associated with the planning process [1]. The idea of developing this app comes from the personal experience of planning trips as a group of friends. Everyone has different date preferences, people post different options on a chat. There is no systematic way to track these options, and the result is that it takes forever to decide. With GroupTripper, instead, users can enter personal date and destination preferences and find common availability. GroupTripper recognizes its users’ needs: having to compare different trip options and overcome the “analysis paralysis”. Users can prepare different trip options, assign research tasks to their friends, and let friends vote to facilitate decision making. This way, GroupTripper allows to complete trip plans faster.

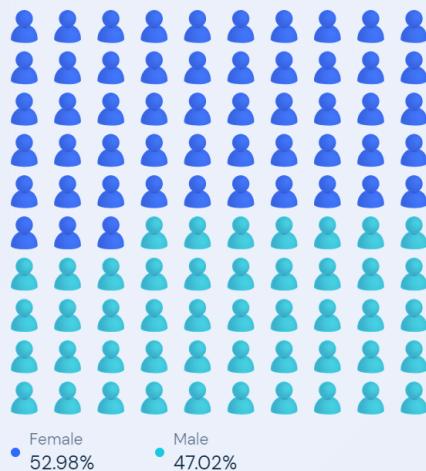
I identify myself and my friends as target users for this application: young adults around the age of 30 who like to travel together and plan their trips in autonomy online. My assumption is reinforced by the audience demographics of existing main competitors: Wanderlog and Troupe [4, 6]. For Wanderlog, the majority of visitors belong to age group 25-34 (32.74%), followed by 35-44 (19.93%), followed by 18-24 (18.25%) [4]. Gender distribution is quite even, with a slight predominance of female users [4]. These trends are shown in Figure 1.

## wanderlog.com Audience Demographics



Audience composition can reveal a site's current market share across various audiences. wanderlog.com's audience is 47.02% male and 52.98% female. The largest age group of visitors are 25 – 34 year olds (Desktop).

### Gender Distribution ⓘ



### Age Distribution ⓘ

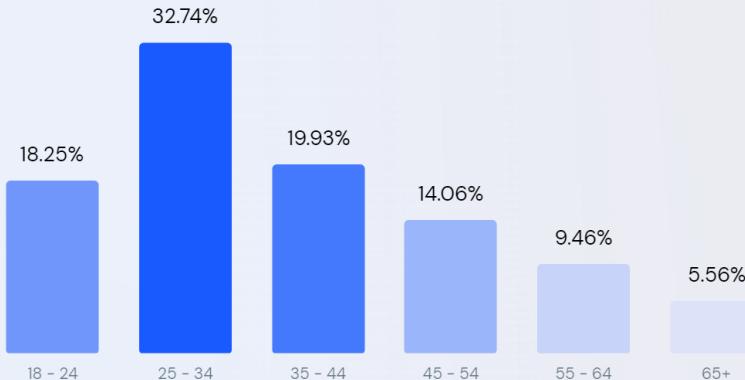


Figure 1 - Wanderlog audience demographics [4]

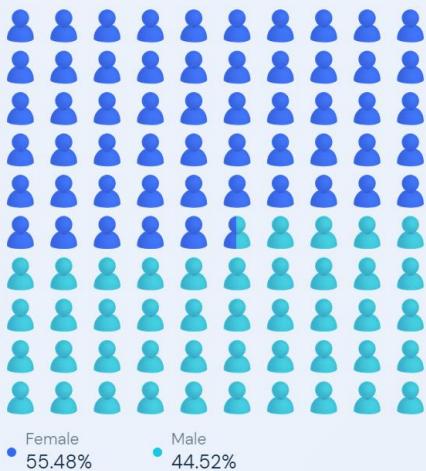
The same trends are confirmed by the audience demographics of Troupe.com reported in Figure 2.

## troupe.com Audience Demographics



Audience composition can reveal a site's current market share across various audiences. troupe.com's audience is 44.52% male and 55.48% female. The largest age group of visitors are 25 – 34 year olds (Desktop).

### Gender Distribution ⓘ



### Age Distribution ⓘ

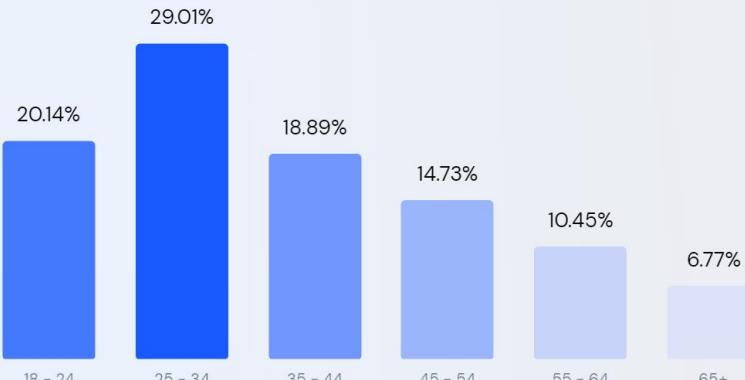


Figure 2 - Troupe.com audience demographics [6]

## Literature review – Market research

It is important to analyze the existing online competition in order to develop a unique market proposition.

To identify potential competitors, I have used the following keywords searches in Google: “trip planning website”, “website for planning trips as a group”, “group trip planning”. Through these keywords I have also found an online article [2], mentioning some of the apps listed in appendix A.

In total, I have identified 10 competitors, listed in order of relevance in the table in Appendix A. The table contains a brief description of the product, as well as strengths and weaknesses, which I gathered after exploring each website. For each solution I noted what features I would like GroupTripper to have, as well as opportunities for improvement.

The most similar solution to my proposition is troupe.com [3], founded in the US in 2019, with 140k total visits in May 2023, coming from English-speaking countries, with most clientele in the age range 25-34 [4]. This is the only website I could find designed with the goal to facilitate planning trips as a group. Although it has some interesting features such as the possibility to do polls and vote on different proposals, the site’s design restrict user activities. Users can choose one date range, one destination, and one stay (no section for transportation), not allowing for a multi-destination trip. I plan to resolve these issues with GroupTripper. I also want to enlarge my customer base by making the site available in languages other than English.

The second-best trip planner website I found is Wanderlog [5], which has great planning features but no options to plan as a group. Wanderlog was also founded in the US in 2019, it has an annual revenue of 1-2M\$, and 5.7 M total visits in May 2023 [6]. Wanderlog’s strengths are the integration with Google API used to log addresses and retrieve pictures for each added location, as well as its map features, which allow to visualize the itinerary and reorganize stops (route optimization is offered with their premium subscription). They also integrate with different third-party APIs to allow in-app booking of hotels, as well as proposing activities by location. Like Troupe, the app is only available in English.

The third best solution is Pilotplans, Canadian web tool with 352k total visits in May 2023 [7]. The tool is very similar to Wanderlog, offering a more intuitive way to create plans and a map view, as well as a live chat. However, it is not well-integrated with third-party APIs, it has no features to plan as a group, and it is also only in English.

Planify [8] stands out by offering the website in 6 different languages and by allowing to download the plan as a PDF. No statistics are available for this site. Planharmony [9] stands out by saving events in a calendar-looking fashion, which also allows to export plans to a preferred calendar. Finally, MiTravel [10] allows to organize trips on a Kanban board.

The rest of the tools listed in Appendix A were not available via web, as they either only offer mobile apps or they are university projects.

In conclusion, I plan to leverage many of the existing features offered by the established apps in this space. In particular, the ability to add different kinds of events (transportation, accommodation, activities, etc.) to a trip via a form, and eventually visualize the locations on a map (after MVP). Things that can be improved in this domain is the possibility to make plans as a group in an intuitive way, targeting a non-English-speaking customer base, as well as ensuring that a website is present and it allows to experience all features before committing to subscribe to GroupTripper. I personally experienced frustration when I tried to compare different tools and discovered that I couldn’t even try out the functionality because either they were available only for IOS, or available only for mobile with no website, or they requested payment right away.

## SWOT analysis

Having identified my competitors and analyzed their solutions, the following are the strengths, weaknesses, opportunities, and threats for GroupTripper.

Strengths	Weaknesses	Opportunities	Threats
<ul style="list-style-type: none"> <li>Value of GroupTripper: a web tool that allows young travelers (23-35) to plan trips as a group.</li> <li>Special functionalities: <ul style="list-style-type: none"> <li>Polls</li> <li>Selective participation on plan items</li> </ul> </li> <li>Usage drivers: <ul style="list-style-type: none"> <li>Preference for booking trips online [1]</li> <li>Analysis paralysis due to too many options [1]. Need to keep track of such options to compare and choose.</li> <li>Only one competitor offering ability to plan as group, poorly designed.</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Time and manpower limitations: only one person working on this project with a limited time span to deliver a working solution.</li> <li>No funding to invest in advertisement to drive adoption of the tool, nor to pay for integration with external APIs (Google, TripAdvisor, etc.)</li> <li>No prior experience in SEO, advertising techniques</li> <li>No time to develop mobile apps</li> </ul>	<ul style="list-style-type: none"> <li>Gain customer trust by providing free access to try out website functionality</li> <li>Tap into non-English-speaking markets</li> <li>Offer a more user friendly experience than the only other tool available to plan trips as a group, by leveraging what works well from other existing products and adding collaborative features</li> <li>Rank high in Google searches by leveraging SEO and good domain name</li> <li>Leveraging available statistics on existing tools to save on advertising (consistent age groups: 25-25, 18-24, 35-44)</li> <li>All competitors are new, started after 2019</li> </ul>	<ul style="list-style-type: none"> <li>GroupTripper's competitors are relatively new, but already have a significant customer base. Additionally, they can leverage startup funding. They are linked in blog posts and they advertise on social media channels. It will be difficult to enter the market. Large investments in advertisement might be necessary</li> </ul>

In summary, although there exist web trip planning tools such as Wanderlog, which already count on 6 million monthly views, only one tool exists offering the possibility to plan as a group: Troupe, with only 140k monthly views and serious design limitations. With GroupTripper I intend to bridge Troupe's limitations and establish as a leader solution in the space of online tools to plan trips as a group. With no funding and time constraints it will be challenging to provide functionality such as API integration, map views, and payment integration. Lack of funding will also limit advertising capabilities, forcing us to purely rely on SEO to rank high on Google search. I will need to research SEO techniques as I have no experience in this. However, I am confident I will set up an MVP outperforming Troupe and that I will be able to successfully further expand on the product adding increasing functionality after launch.

# Design

## Requirements and acceptance criteria

In order for GroupTripper to become the leading web tool to plan trips as a group, I defined a list of functional requirements and acceptance criteria based on the market research and SWOT analysis reported above. The table in Appendix B reports the features I plan to implement within GroupTripper's MVP and it describes the goal, user story, requirement, and priority of each feature. Features with high priority constitute the acceptance criteria for the MVP.

## Use case diagrams

Appendix I contains three UML use-case diagrams to display the behavior of GroupTripper in relation to its three types of users: authenticated new users, unauthenticated users that already have an account, and unauthenticated ones.

## Prototypes

I followed an iterative approach to design the prototypes. Starting from the functional requirements (with high priority i.e. success criteria for MVP) reported in Appendix B, I have drafted a first series of medium-fidelity prototypes, which are displayed in Appendix C.1. I then refined these prototypes (Appendix C.2), created a Google Forms survey [11] and shared the second iteration prototypes with a group of friends that represent the target audience of GroupTripper, in order to get feedback and recommendations. Although I only received 2 responses, I managed to incorporate those recommendations into the third iteration of the prototypes that are shown in appendix C.3. I decided not to create high fidelity prototypes for this project, due to time constraint and because in my opinion medium-fidelity ones are enough to reference in the implementation phase and are thus a sufficient time-effective effort. Instead, it takes a lot more time to develop high-fidelity prototypes and due to limitations of the most popular software to produce them – Figma – it is not possible to translate design into HTML and CSS automatically without paying. I have no budget dedicated to developing this product and hence creating detailed designs on Figma would only mean having to spend double time on designing an esthetically pleasing UI (before replicating the effort with HTML and CSS). Medium-fidelity prototypes ensure that all functionality has been thought of. As for esthetics, I plan to make the website as simple as possible prioritizing accessibility constraints. Light minimalistic background, with properly contrasted dark text. You can find more details on this in the Front-End section of this report below.

## Technical infrastructure

### Back-end

When it comes to choosing how to deploy a REST API or web app there are multiple options to be weighted based on one's priorities. My priorities are: first, I want to be able to deploy my application fast, without having to take care of servers and networking configurations, and without having to take care of scaling up resources manually when the user base grows. Second, I want to do so in the cheapest way possible. Finally, the infrastructure needs to be able to accommodate the features that my app requires.

There are different cloud services available, which ensure scalability and require different degrees of manual configurations to do. I will compare BaaS and IaaS solutions and explain why I ultimately chose BaaS.

## *BaaS - Google vs AWS*

“Backend as a Service (BaaS) is [...] a modern cloud computing type that is useful to build mobile applications rapidly. Startups, SMEs and large enterprises prefer to go with BaaS solutions because of their low cost, speedy development and other benefits.” [12].

Two common BaaS solutions in 2023 are Firebase (Google) and Amplify (AWS) [12]. These are both “excellent choices for serverless architecture apps in Web Development”[19].

AWS Amplify facilitates back-end and front-end development and offers web hosting for free for a year, then pay as you go based on resources consumption [13]. The main differences between Amplify and Firebase “rely on the database implementation, APIs protocols, and cloud services each platform runs on top”[17]. “Amplify is more complicated and has a higher learning curve than Firebase. It also needs additional setup and tuning.” [14]. This makes Firebase more ideal for small projects where time to market is critical [18]. Firebase offers an unlimited free tier [16, 20], as well as free easy authentication integrating with third-party services (Google, Facebook etc.) [15].

Given the above information, Firebase seems more suited to the requirements of this project and it is therefore my preferred choice over Amplify.

## *BaaS vs. IaaS*

“Infrastructure as a service, or IaaS, delivers on-demand infrastructure resources to organizations via the cloud, such as compute, storage, networking, and virtualization. Customers don’t have to manage, maintain, or update their own data center infrastructure, but are responsible for the operating system, middleware, virtual machines, and any apps or data.” [21]. App engine is Google’s IaaS [21], while AWS offers equivalent services with Beanstalk [23, 24], although at a slightly higher price [28].

Essentially, the main difference between Firebase and App Engine is the lack or presence of a backend respectively [22]. For example, with App Engine one can deploy applications built with frameworks such as Django and decide to set up complex architectures to handle custom data storage and processing solutions (NoSQL vs. relational vs. Data lake, batch vs. streaming). Example of companies using App Engine are Snapchat and Dialpad (enterprise cloud phone system), both needing to process big data (streaming, large amount, unstructured data) [25].

On the other hand, Firebase is used by companies such as Alibaba, The New York Times, The Economist [26], Lyft, Duolingo, and Venmo [27], where data processing is more straightforward. Google App Engine offers standard and flexible plans [29], the flexible plan allowing for WebSockets and demanding at least 1 server stays working at all times.

## *Does GroupTripper require a complex back-end solution?*

The only use case for my application to require a backend beyond a NoSQL database is if I wanted to build live chat functionality. Although this is one of the requested items within the project template to get full marks, I am not sure it makes sense to stretch my requirements and set up a more expensive infrastructure [30] to implement a feature that is actually not necessary for my application (and which you can check I have already correctly implemented in the Advanced Web Development final project [31]). While building an API with a framework like Django is faster than writing all the backend logic manually [34], the time saved in configuring the backend justifies in my view going with Node.js and Firebase, as reinforced by many views on a related Quora post [35].

Having to write each database call as opposed to using Django's ORM also comes with flexibility [34]. As for security, we can follow best practices to minimize the risks to incur cross-site scripting, cross-site forgery requests, code injections, and other common attacks [32, 33].

#### *The chosen solution*

Appendix J displays the technical diagram of my chosen solution. I will use Firebase to host the website, leveraging Firebase authentication, and the No-SQL Firestore database (no real-time needed). I will use cloud functions to store the application code, which will be written in JavaScript using Express: "a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications." [36]. I will also make use of Ajax to make asynchronous calls to the Firestore database [40] for certain features. Finally, I will need to send emails to users requesting to sign up or to send them notifications. To do this, I will explore solutions mentioned on StackOverflow [41], which reference an official Firebase Github solution to send confirmation emails with different email sending platforms [42]. Appendix D contains a price comparison of different email platforms. I chose to use Google's SendGrid as it's the cheapest. In order to allow users to select places with autocomplete from a drop-down menu when creating itineraries, I have compared different options and their pricing in appendix E and decided to use MapQuest [57].

#### Front-end

For the front-end I am going to use Bootstrap components and EJS templating. I will use palettes with accessible color contrast [37] and test color accessibility using Wave [39] once my website is hosted, following color accessibility best practices [38].

#### API design

Appendix K displays the REST API end points I designed, which leverage HTTP methods supported by all modern browsers [43] and follows REST API design best practices [48]. The implementation of these calls is closely related to the considerations reported in the Data store design section. I will use content negotiation to have each end point accept / return JSON or HTML, in order to improve SEO [52].

#### Data store design

"When using Firestore, the mechanism for billing is related to the number of reads, writes, and deletes we perform." [44]. Therefore, I will design my solution in a way such as to minimize reading and writing, while considering Firebase architecture [46] limits e.g. maximum size for a document is 1MiB [45]. I will also have my application maximize caching database queries where it makes sense [47]. Appendix F contains the design for the two Firestore collections I will create: "users" and "trips", done considering data types available in Firestore [58]. Documents will be updated using Firestore transactions [59]. Database permissions will be set using Firestore security rules [60].

#### Realtime web technologies

I plan to use Firebase snapshot listeners on the client side, to ensure database updates are received in a real-time fashion by all users [120, 117, 119]. This is an alternative technology to Web sockets, which is explained in detail on Google's documentation [116].

## Monetization

I plan to have a free app version and a Pro subscription version. The free app version will use Google Adsense [64] to cover what I expect to be minimal infrastructure costs. Users will be able to switch to the Pro version by paying 5.99 € per month (automatically renewed) via an integrated payment method on the website. Appendix G displays a comparison of 2 Payment platforms: Stripe and Square. I chose to use Stripe due to same pricing as Square but more included support. Appendix H shows the features that will be available only with a Pro subscription.

## Legal considerations

I plan to consider legal obligations under privacy, consumer protection, copyright, and defamation law [72].

## Search Engine Optimization

I intend to follow SEO best practices such as choosing a meaningful domain name, using the right keywords (repeated at the top of the page [74]), ensuring all links from my website work [73]. Appendix M contains a full list of SEO techniques I am going to use.

## Evaluation plan

In order to ensure I develop a successful and functioning application, I will:

- 1) Implement Unit tests with good test coverage (white box testing) using Node.js Test runner [110] and node-mocks-http [112].
- 2) Black box testing: I will ensure that all functionality listed as acceptance criteria (Appendix B, priority high) works as expected, manually.
- 3) Run vulnerability checks on my live application using two or more web security scanners [70, 71].
- 4) User testing: after the MVP is ready, I will invite people to test it and ask to answer a questionnaire that will include questions recommended by usability testing experts [69]. I plan to get at least 20 responses to this questionnaire.

## Development plan & timelines

Appendix L.1 shows a Gantt chart containing a detailed description of milestones, tasks, and timelines, which I plan to follow in order to complete this project on time. Milestones include infrastructure setup, security aspects, pages development, and report writing. Hard deadlines are shown in red. Orange tasks are yet to be implemented, green tasks are those that have been completed, yellow means in progress. I plan to update this chart every week. I have assigned timelines to milestones and assigned a period of one week to most milestones, depending on the complexity of the tasks within. I will work in an Agile fashion, with sprint periods of one week. Each week I plan to work on the assigned tasks and reassess priorities as I go along in case I encounter unforeseen difficulties and need to alter the work plan, or conversely in the optimistic case in which I progress faster than expected. I do consider these to be realistic timelines, based on similar work I have completed in about the same amount of time for other classes of this degree (Advanced Web Development, Web Development, Agile Software Projects) and my progress so far. Appendix L.2 contains a critical path analysis table, which references each milestone of the Gantt chart and indicates each milestone's dependency. For example, we can see that milestone 3 (authentication) can only be started once milestone 2 (infrastructure setup) is complete and it will take 9 days of development.

## Implementation



Figure 3 - GroupTripper logo, made with Inkscape [81, 82, 83].

GroupTripper is a web application that requires users to register accounts to start collaborating on the platform in creating trip plans. As such, the most important features are: setting up the infrastructure, including database and hosting, setting up a secure authentication mechanism, ensuring that the front-end can effectively communicate with the backend, saving and retrieving information to/from the Firestore database with appropriate API calls, ensuring the correct behavior of routers via unit tests, ensuring that users can communicate in real-time, ensuring that appropriate permission rules are implemented to control user access to the database, and finally that all features mentioned in appendix S are implemented and function correctly. I have implemented all the above and my website is accessible at <https://grouptripper-3c7f1.web.app/>. Currently, registration required a beta test code: “girotondo”. For the moment I am not allowing Google to index the website [105]. Once the product is fully ready, I will remove the “<meta name="robots" content="noindex,nofollow" />” from /functions/views/partials/sharedHead.ejs

I have designed the website logo (Figure 3) with Inkscape by following a few Youtube tutorials [81, 82, 83]. The project is stored in a public [Github repository](#) [122]. Appendix R further proves usage of Git.

### Entry point and Routers

The entry point of the application is functions/index.js [142]. Here, Firebase is initialized and Express [36] is also initialized. At lines 72-79 all the routers are called, enabling users to call 8 main routes, with additional paths and relative HTTP request methods specified in the routers themselves. Each router logic is specified in a dedicated file [143]. For example, the settings router [144] defines 2 API end points: GET /settings and POST /settings. When a request returns status code 404, a dedicated page is rendered – implemented in index.js, lines 81-96 [142]. Routers can check whether users are authenticated and display information accordingly. They can interact with the Admin SDK and connect to the database, then use this content to fill EJS templates that are returned to clients. For example, the settings router end point GET /settings checks that users are authenticated at line 19, then interacts with the database at line 20 to get or create user settings, and finally renders a template filled with the information retrieved from the database at line 31 [144].

### Ajax calls

Some of the API endpoints are called from the frontend via Ajax calls. An example of this can be seen when users want to delete a poll. Within tripPolls.js [145], line 324, I define that when the user clicks on the “delete poll” button, they are prompted with a pop-up asking to confirm the poll deletion. If the user confirms, function deletePoll is called. The function is defined at line 138 [145] and it contains an Ajax call to /trips/<trip ID>/polls/<event ID> with method DELETE, which has the effect of deleting the poll from the database.

### Hosting and Firestore database set-up

To set up hosting and Firestore database, I have followed the instructions to set up web projects on Google’s documentation [84]. However, this was not a straightforward process, as I had a few issues related to the cloud region as well as errors thrown by the Firebase CLI during set up, so I’ve had to repeat the process manually a couple of times before everything worked. I have documented all the set-up steps within my Github repository’s “DEVELOPMENT.md” file, located at the root of the project.

### Authentication

By far the hardest feature to implement has been authentication. My application consists of a Node.js backend API, which fills EJS templates upon request and sends them to clients, and a simple JS + HTML + CSS front-end

from which clients make HTTP requests and navigate to different pages, being served different content. Mine is thus not a single page application, although in some instances, API endpoints are called in the background with AJAX calls (e.g. saving notification setting when user clicks on toggle on the UI).

I initially implemented authentication using Firebase client SDK for the web [85] instructions. This was quite straightforward. However, when it came to test the application I soon realized I was being asked to login at almost all page refreshes. Clearly something was wrong.

I then looked into how to persist user authentication [86] and tried to implement the proposed solution. However, HTTP requests started to time out. I initially tried to implement manual ways to persist authentication on the backend with Node [87], but that seemed like a very unsafe and hackey method to me. It worked, but it meant saving users authentication on local files on the server.

At this point I was a bit anxious that I would not be able to meet the deadlines, not being sure I'd manage to find a solution. I continued searching and found out that Google Firebase actually has 2 SDKs: client and admin. I found that the proper way [88, 91, 92, 93] to implement authentication in this front-end + backend scenario with Firebase is to use client SDK on the front-end, set up session cookies and have the backend enabling and validating sessions via the admin SDK. Users must log-in i.e. initiate sessions from the front-end using the client SDK [89, 95]. When the user logs in, they send an HTTP POST request to the /auth/sessionLogin end point. This request is prepared by the client SDK and it contains an ID token [90], as well as a csrfToken. Upon receiving this request, the back-end's admin SDK verifies the ID token and creates a session cookie that will be valid for the next 8 hours (my custom implementation) [94]. The session cookie is attached to the request response and returned to the client that way. At this point, all subsequent client requests will contain the session cookie. The back-end verified the integrity of all requests by checking such session cookie to determine what user is currently logged in and to display/return content accordingly. While some API calls don't require user authentication, others require to return data that only one specific user can have access to.

At this point everything seemed to work when running the application on localhost with the Firebase emulator [96]. However, once I deployed the app to the cloud, suddenly sessions were not working. Upon further deep diving on the issue, I found that when running on Firebase hosting, cookies can only be stored under the name of “\_\_session” [97]. “When using Firebase Hosting together with Cloud Functions or Cloud Run, cookies are generally stripped from incoming requests. This is necessary to allow for efficient CDN cache behavior. Only the specially-named \_\_session cookie is permitted to pass through to the execution of your app.” [97]. Finally, authentication did work as expected. Snippets of authentication implementation are available in appendix Q.

## Bootstrap 5

The process of authentication is one of the many occasions in which I like to display HTTP request responses to users, to communicate with user-friendly messages the outcome of users' actions.

The front-end is implemented in vanilla JS, with EJS templates being rendered to HTML with CSS. I make use of Bootstrap 5 [99] to quickly create modals, navigation bar components, buttons etc. One minor issue I stumbled upon is the difference between bootstrap 4 and Bootstrap 5 when it comes to implementing certain functionality like modals. In particular, I've experienced the issue that modals would not work. Once aware of this difference, the problem was straightforward to fix: I just needed to modify a few HTML attributes for the modals to work [100].

I also used one of Bootstrap5's login forms [101] to display during the login, signup, and password reset phases.

## [Input validation and sanitization](#)

On the front-end, input validation is enforced making use of regular expressions and other checks. Custom error messages are displayed to users explaining required rules (`/functions/views/authentication.ejs` in the repository) [102]. On the backend, I found [103] that Express comes with its Express-validator [104] module so I decided to use this. An example of sanitization can be found in `/functions/routers/authentication.js` at the beginning of the POST `/signup` method. Appendix P contains examples of input validation and sanitization implementations.

## [Database interactions and security](#)

API calls are handled on the backend, making mixed use of client and admin SDK. All calls that require user authentication are handled via admin SDK, which uses secret credentials stored as configuration file in the repository. After the evaluation of this project is concluded, I will rotate credentials and make the repository private or find another way to store this sensitive information. Calls made with the client SDK, on the other hand, leverage other configurations keys, which can be made public to users. Clients make calls using the Firebase SDK when they need to login and start a session (constraint explained above), as well as when real-time communication is required. Access to the Firebase database is controlled by security rules [106]. The security rules currently implemented are available in the repository [129] and ensure that users can access only data that is relevant to them, as demonstrated in a dedicated article [118].

Most database interactions are handled by the admin SDK on the back-end, ensuring that users are authenticated (session cookies, explained above). I have implemented all required operations and queries to create and interact with two collections: users and trips, described in Appendix F. Basic operation [130] and specific queries [131] are implemented within the repository in `/functions/db/` and are also shown in Appendix O.

## [Realtime technologies](#)

The web development collaboration web application template mentions that outstanding student projects should make use of realtime web technologies such as Websockets.

Manually implementing Websockets is not recommended / supported with Firebase projects [113]. The supported alternative that implements WebSockets is to use the Firebase realtime database [114], which can be however quite expensive [115]. However, upon further investigation it turns out [116] that even the standard Firestore database allows near-to-real-time (low-latency) querying at a lower price. “Real-time queries, also called snapshot listeners, let the app listen to changes in the database and get low-latency notifications as soon as the data changes. An app can get the same result by periodically polling the database for updates, but it's often slower, more expensive, and requires more code.” [116]. I have implemented real-time functionality in my application in order for users to comment on plans and polls, see new plan updates, and receive notifications in real-time. To implement these functionalities, I have therefore implemented snapshot listeners on the client side, to ensure updates are received in a real-time fashion by all users [120, 117, 119]. For details of how snapshot listeners work, you can reference Google’s documentation [116]. For example, a snapshot listener is implemented to get poll events in real-time from the application front-end [132].

## [Useful libraries used](#)

In order to let users upload and resize their profile picture, I used Croppie [134, 135, 136]. Profile pictures are directly saved as base64 encoded strings in Firestore [137, 138]. To implement searchable select menus with custom options I leveraged the Tom-select JS library [139, 140]. I used these for example to let trip owners choose who they can add to a trip poll, pre-populating the menu only with the user’s friends [141].

## Evaluation

I implemented all features I planned to implement in my Gantt except 2 ones (Google Adsense and payment integration with Stripe). All functional requirements specified in Appendix B with Priority ‘High’ i.e. all acceptance criteria defined in the planning section of this report are complete. I am overall satisfied with the implementation. I have done all that was in my power, given the time constraints, to insist on reasonably high standards during the research, design, and implementation phases. My application is serverless (thus scalable) and secure (session cookies, Firebase hosting with zero-configuration SSL [121], database rules correctly configured). All routes implemented have unit tests and behave correctly.

I also implemented real-time updates for clients (e.g. notifications are received by all relevant clients the moment they are sent). To do this, I leveraged Firebase onSnapshot listeners [116]. The implementation of Snapshot listeners is proprietary to Google [128], but it seems [128] that the technology behind it is an implementation of the WebSocket protocol, which defaults to long polling when the first option is not supported by the browser. Plausibly, the client opens a WebSocket channel to the Firebase server and communicates live updates through it. Due to time constraints some aspects of my implementation of real-time updates could be improved. For example, when multiple users comment on the same event in a trip and a new comment is posted, the comments section is closed automatically (and its content updated). The user has to reopen the section to see the update. I plan to implement a better implementation for these type of features after the MVP launch.

As described in the design section of this report, my formal evaluation plan comprises 4 steps: unit tests, black box testing, security, and usability testing. The following sections cover these points.

### Whitebox testing - Unit tests

Before this project I had never implemented unit tests in JavaScript. I did some research and the most common libraries to write unit tests in JS seem to be Mocha and Jest [108]. I later found out [109] that in 2022 Node.js implemented their own integrated testing framework: Node.js test runner [110, 111]. I decided to use Node.js test runner in combination with node-mocks-http, a library to test Express routing functions [112]. I eventually managed to correctly set up tests folders and package.json. Unit tests can now be run with the ‘npm test’ command, from the project root folder. Unit tests configurations, examples, and outputs are available in Appendix N. The first picture in appendix N.1 shows the configurations needed to run ‘npm test’, while the second figure in appendix N.1 shows the required dependency node-mocks-http. The image in appendix N.2 shows that the folder structure of the tests mirrors the folder structure of the project. For each router there is a test folder containing related unit tests. I have implemented unit tests also for two utils functions: authentication and time. The image in appendix N.3 shows an example of unit test for the home router. Here, the first unit test checks that when a user is logged in, visiting the “/” route redirects them to the /trips page. The real behavior of the routers in this application is that a function called verifySessionCookie is called to ensure that the user who makes a request has a valid session i.e. decrypts session cookies and returns user session information (who is logged in, since when, etc.). To mimic user authentication, function verifySessionCookie is mocked, returning fake user-related details (l. 13). At line 31, a mock GET request is emulated, containing cookies with the “\_\_session” field. When the response event ends, lines 41 to 47 check that the response status code is 302 (redirect code), and that when redirecting the session cookies are passed along. Unit tests cover all scenarios possible in the routes logic and have a very similar structure throughout the code. Appendix N.4 shows the results of the unit tests: all 23 tests pass. Although all routes have unit tests, unit tests coverage could be increased and the existing unit tests could also be more detailed. Due to time constraints I was not able to precisely mock all

database interactions and only tested the generic functionality of routers such as status codes, rendering, and redirects.

## Blackbox testing

In order to ensure that all the website functionality behaves as expected, I have thoroughly performed all operations that users will perform and checked that they work correctly. In appendix S, I report a comprehensive list of all features implemented. This corresponds to the list of functional requirements specified in Appendix B with Priority ‘High’, which is the list of acceptance criteria defined in the planning section of this report. I describe the testing procedure followed, as well as the test results (pass / fail). All acceptance criteria are correctly implemented and thus all tests until feature 21 pass. Due to time constraints I was not able to implement features 22 to 24.

## Vulnerability checks

As mentioned in the Design - Evaluation plan section, I have referred to Kumar [2023]’s third option – i.e. <https://hostedscan.com/> to scan my website for security vulnerabilities. This in turn leverages OpenVas, Owasp\_ZAP, and Nmap. OpenVas is “one of the recent and popular open-source tools providing vulnerability scanning and vulnerability management options.” [71]. “The scanning engine of the tool is regularly updated with the Network Vulnerability tests.” [71] and “The tool is equipped with identifying security issues in the network servers and devices.” [71]. In 14 minutes, Hostedscan.com [123] completed the checks. All resulting reports in HTML format are available within Appendix T. The first scanning found 5 medium-security risks and 9 low ones.

One of the reported vulnerabilities regarded setting the correct cross-origin-resource policy, so that the session cookies, which contain confidential information, are not sent to third party-sites. I was able to solve this by implementing [a small change to fix a syntax error](#) [124] for all pages that require authentication. Personally, I consider the clearest report to be the one generated by the OWASP scanner, which classifies vulnerabilities by severity and explains in simple words, well-formatted what is the issue. Unfortunately, although I have tried to resolve more issues related to headers by using the helmet library [125] as suggested by Edwin [126], a second scan gave more or less the same results (only 2 solved risks), as shows in Appendix T.5. Additionally, by addressing security vulnerabilities, some of the functionality of the website stopped working, so I have had to roll back security-related changes. Due to time constraints, I will leave the resolution of these risks to be addressed as future improvement, after launching this MVP.

## User testing

I have created a [survey](#) with Google Forms to evaluate GroupTripper’s usability, referencing 7 of the questions recommended by UX-design experts [69]. I shared this survey via Email with 20 people (work colleagues and friends corresponding to the target demographic (25-34 year old [4, 6])). Responses highlighted some bugs and improvement suggestions which I report in Table 1.

*Table 1 - Bug report and improvement suggestions*

N.	Suggestion / Issue
1	From mobile, when no notifications present, the menu still presents an empty list
2	When signing up, the username field does not accept numbers
3	The email that is sent confirming user registration does not contain a link to directly go back to login

4	Trip common dates availabilities are not visible enough. They should be highlighted
5	From mobile, when navigating backwards in pages, bugs emerge in the process of adding friends to trips
6	When the owner of a trip finalizes common dates, all trip members should be notified.
7	Make it clear what fields are mandatory to fill in forms by marking them with a star
8	Don't notify self when creating an event
9	Notify trip owner when all participants decide whether to join the trip or not
10	Bug when seeing comments – need to click twice
11	Notify event owner when a comment is added
12	From mobile: notifications settings events are fired even if a user scrolls and to do so clicks on the row where a switch is present. Events should only be fired when users click exactly on the notification setting switch (stop event propagation).
13	Password form in login and signup should have an 'eye' icon on the side that allows to briefly display it.

Unfortunately, due to time constraints, I cannot address these items before the submission for this assignment. I do, however, plan to address them at a later stage so that my website can be used comfortably by all users, including myself.

### Planning vs. execution evaluation

As mentioned in the black box testing session above, I could implement all main functionality for GroupTripper as defined in the planning phase of the project. However, due to time constraints I could not tackle some of the challenging features I wanted to explore. Namely, in the initial Gantt chart displayed in Appendix L.1 I planned to dedicate time to integrate Google Adsense and Stripe for payments. Unfortunately, implementing the main functionality took longer than originally planned, so I will integrate advertisement and payments at a later stage. I have also had to rearrange the order of implementation of the existing features, as the initial critical path analysis turned out to be inaccurate. I realized during the implementation that, for example, in order to implement the trip participants page, I first needed to implement user profiles and the friendship model. Only then any user could add other users to a trip. I thus altered the original Gantt chart to reflect the actual progress I made until completion. This is available in Appendix L.3.

### Conclusion

The objective of my final project was to build a collaboration web application that allows to plan and track trips as a group more effectively: GroupTripper. I wanted to demonstrate my ability to build a modern and secure web application that leverages cloud technology, integrates with third party APIs, optimizes SEO, and considers monetization strategies including hosting, advertisements, and allowing in-app payments via Stripe. I wanted to developed a POC of a product that could be launched successfully on the market and be a potential source of income, offering a better user experience than that offered by my major competitor already on the market: Troupe [3].

I have only partially succeeded in achieving my original objectives: I have developed a fully functional POC for GroupTripper, deployed and available at <https://grouptripper-3c7f1.web.app>. Based on the results of my security tests, the application is secure (no high risks reported). I used Google cloud technologies to implement the website's back-end as well as for hosting. I have integrated a third-party API, Pexels, to dynamically associate pictures to trip titles, I have implemented SEO considerations (currently the site is not indexed by search engines

as I want it to be only when fully ready to use), and it integrates all main functionality offered by Troupe and that I planned to implement for the POC. Although I have thoroughly considered monetization strategies in the planning phase, due to time constraints I was not able to implement this aspect.

The application currently allows users in possession of a beta test code to correctly register to the website and collaborate in real time on the creation of trips.

Much development work is still required before this product can be successfully launched on the market. First, all bugs reported so far need to be solved, and further usability testing needs to be carried out especially on mobile devices. Second, database interactions can be optimized to reduce calls to the minimum in order to avoid spikes in costs once many users actively interact with the product. Database queries also need to be refactored and rewritten as transactions [133] as currently sequences of operations can't be performed as a unit. This is a crucial consideration that needs to be addressed before implementing payment features. More time needs to be dedicated to solving all security concerns highlighted by the security reports, and unit tests need to be expanded to thoroughly comprise interactions with databases and details of template rendering. Real-time update features need to be improved so as not to disrupt user activities already in progress (e.g. if a user is writing a comment and another user posts an update, currently the comments section gets closed and the active user needs to restart writing the comment from scratch).

Before launch, I would need to buy a domain name for the website that can be easily found from web searches. Only when all these aspects are address, will I be able to allow search engines like Google to index the website and let any user start using the platform. Before allowing Google to index my website, I would need to ensure all SEO considerations are correctly applied (e.g. only public links can be followed – no pager requiring authentication should be indexed). Once all this is done I would need to start advertising the website, connecting to bloggers that reviewed similar applications to also try out GroupTripper and review it. This will allow more and more websites to link to GroupTripper, which will appear in the top results for its relevant key words and be successfully found. Finally, in order to attract a diverse audience, the website will need to be available in different languages.

Unfortunately, time constraints were the most important limitation of this project as mentioned in the SWOT analysis in the literature review section of this report. Other limitations I mentioned then and are still relevant are lack of funding for this project, no prior experience in SEO, and no time to develop a mobile application on top of the website.

I do consider this to have been a successful experience as I have managed to research all that is needed to launch a professional website. I have explored the latest trends in terms of cloud technologies, compared options available on the market to send emails, implement geo-location, search for pictures. I have developed a functioning POC that showcases my ability to design, plan, and implement a project in a short amount of time, namely 3 months, mainly only working on weekends as I have a demanding full-time job on the side. I am proud of what I was able to achieve and hope to have the time to address all the improvements I mentioned above to one day see this product being used at least in my circle of friends and possibly by a larger pool of users.

## References

- [1] TravelPerk. 2022. 60+ online travel booking statistics & trends. Retrieved May 18, 2023 from <https://www.travelperk.com/blog/online-travel-booking-statistics/>
- [2] Patkar, Mihir. 2023. 5 Group Travel Planning Apps to Make Trips With Friends and Family Easier. Retrieved June 19, 2023 from <https://www.makeuseof.com/group-travel-planning-apps/>
- [3] Troupe. Retrieved June 19, 2023 from <https://app.troupe.com/>.
- [4] Similarweb. Troupe. Retrieved June 19, 2023 from <https://www.similarweb.com/website/troupe.com/#overview>
- [5] Wanderlog. Retrieved June 19, 2023 from <https://wanderlog.com/>
- [6] Similarweb. Wanderlog. Retrieved June 19, 2023 from <https://www.similarweb.com/website/wanderlog.com/#ranking>
- [7] Pilotplans. Retrieved June 19, 2023 from <https://www.pilotplans.com/>
- [8] Planify. Retrieved June 19, 2023 from <https://planify.io/>
- [9] Planharmony. Retrieved June 19, 2023 from <https://www.planharmony.com/>
- [10] MiTravel Retrieved June 19, 2023 from <https://mitravelapp.com/>
- [11] Google Forms survey to evaluate first round of medium fidelity prototypes. Retrieved June 19, 2023 from [https://docs.google.com/forms/d/1uYtOSe0qjQ9ILG7yugXDw2a9vic\\_88VFcM1stR\\_32dI/edit#responses](https://docs.google.com/forms/d/1uYtOSe0qjQ9ILG7yugXDw2a9vic_88VFcM1stR_32dI/edit#responses)
- [12] Back4App. 2023. Top MBaaS Providers in 2023. Retrieved June 19, 2023 from <https://blog.back4app.com/mbaas-providers/>
- [13] AWS. 2023. AWS Amplify pricing. Retrieved June 19, 2023 from [https://aws.amazon.com/amplify/pricing/?trk=d3adb855-b91b-4e74-8308-5e9f08e34ed2&sc\\_channel=ps&ef\\_id=Cj0KCQjw7PCjBhDwARIsANO7CgIHASwn4PwmKoTOrJa4oDvzMJwm7IrrMGMdMS2IE54WuTkRjE3Jet8aAuazEALw\\_wcB:G:s&s\\_kwcid=AL!4422!3!647302000972!e!!g!!aws%20amplify%20pricing!19621396985!145160419149](https://aws.amazon.com/amplify/pricing/?trk=d3adb855-b91b-4e74-8308-5e9f08e34ed2&sc_channel=ps&ef_id=Cj0KCQjw7PCjBhDwARIsANO7CgIHASwn4PwmKoTOrJa4oDvzMJwm7IrrMGMdMS2IE54WuTkRjE3Jet8aAuazEALw_wcB:G:s&s_kwcid=AL!4422!3!647302000972!e!!g!!aws%20amplify%20pricing!19621396985!145160419149)
- [14] Muzammil, K. 2023. Amplify vs. Firebase: Which Is Better in 2023. Retrieved June 19, 2023 from <https://www.aalpha.net/articles/amplify-vs-firebase-which-is-better/>
- [15] Titus Efferian. 2023. Amplify vs. Firebase: A Head-to-Head Comparison With My Real Product. Retrieved June 19, 2023 from <https://betterprogramming.pub/amplify-vs-firebase-a-head-to-head-comparison-with-a-real-product-b6fd76058416>
- [16] Mark Basson. 2021. Firebase vs AWS Amplify. Retrieved June 19, 2023 from <https://medium.com/geekculture/firebase-vs-aws-amplify-2286389c63cf>
- [17] Mariana Clark. AWS Amplify vs Firebase – Which is better? Retrieved June 19, 2023 from <https://blog.back4app.com/aws-amplify-vs-google-firebase/>
- [18] BairesDev Editorial Team. Amplify vs. Firebase: Which Is Best Suited for Your Project? Retrieved June 19, 2023 from <https://www.bairesdev.com/blog/amplify-vs-firebase-which-one-is-best/>
- [19] Parth Thakkar. 2022. Firebase Vs AWS: Which One to Choose in 2022? Retrieved June 19, 2023 from <https://medium.com/mqos-technologies/firebase-vs-aws-which-one-to-choose-in-2022-dce230ab44dd>
- [20] Firebase pricing. Retrieved June 19, 2023 from <https://firebase.google.com/pricing>
- [21] PaaS vs. IaaS vs. SaaS vs. CaaS: How are they different? Retrieved June 19, 2023 from <https://cloud.google.com/learn/paas-vs-iaas-vs-saas>
- [22] Jessica Clark. Google App Engine vs Firebase. Retrieved June 19, 2023 from <https://blog.back4app.com/google-app-engine-vs-firebase/>

- [23] AWS. AWS Elastic Beanstalk. Retrieved June 19, 2023 from <https://aws.amazon.com/elasticbeanstalk/>
- [24] Kurt Marko. 2021. PaaS showdown: AWS Elastic Beanstalk vs. Google App Engine Retrieved June 19, 2023 from <https://www.techtarget.com/searchcloudcomputing/tip/PaaS-showdown-AWS-Elastic-Beanstalk-vs-Google-App-Engine>
- [25] Tess Townsend. 2017. This is what Snap is paying Google \$2 billion for. Retrieved June 19, 2023 from <https://www.vox.com/2017/3/1/14661126/snap-snapchat-ipo-spending-2-billion-google-cloud>
- [26] Jessica Clark. Top 10 Companies Using Firebase. Retrieved June 19, 2023 from <https://blog.back4app.com/companies-using-firebase/>
- [27] Firebase. Trusted by the largest apps and games. Retrieved June 19, 2023 from <https://firebase.google.com/#:~:text=Trusted%20by%20the%20largest%20apps,Firebase%20to%20ship%20their%20apps>
- [28] Joseph Holbrook. Summing Up Amazon Elastic Beanstalk Vs. Google Cloud App Engine. Retrieved June 19, 2023 from <https://thegcpgrus.com/summing-up-amazon-elastic-beanstalk-vs-google-cloud-app-engine/#:~:text=Pricing,as%20VMS%2C%20storage%20and%20networking>
- [29] Choose an App Engine environment Retrieved June 19, 2023 from <https://cloud.google.com/appengine/docs/the-appengine-environments>
- [30] Przemek. 2020. Django: How much it costs to host a small project on App Engine. Retrieved June 19, 2023 from <https://pnote.eu/notes/django-app-engine-cost/>
- [31] Advanced Web Development final project by Susanna Ventafridda. WebSockets usage. Retrieved June 19, 2023 from [https://github.com/SusyVenta/AdvancedWebDev/blob/main/final\\_project/BuzzXchange/social\\_network/consumers.py](https://github.com/SusyVenta/AdvancedWebDev/blob/main/final_project/BuzzXchange/social_network/consumers.py)
- [32] Top 10 Node.js Security Best Practices Retrieved June 19, 2023 from <https://snyk.io/learn/nodejs-security-best-practice/>
- [33] Dawid Ziolkowski. 2021. Top 11 Node.js security best practices. Retrieved June 19, 2023 from <https://blog.sqreen.com/nodejs-security-best-practices/>
- [34] 2023. Django vs Node.js: Difference between Django and Node.js. Retrieved June 19, 2023 from <https://intellipaat.com/blog/django-vs-nodejs-difference/?US>
- [35] Quora. What is your opinion on Django vs Firebase as a mobile app backend? Retrieved June 19, 2023 from <https://www.quora.com/What-is-your-opinion-on-Django-vs-Firebase-as-a-mobile-app-backend>
- [36] Express Retrieved June 19, 2023 from <https://expressjs.com/>
- [37] Accessible Colour Contrast. Retrieved June 19, 2023 from [http://www.sussex.ac.uk/tel/resource/tel\\_website/accessiblecontrast/?q=FFFFFF~003b49~1d4289~94a596~e56db1~d3273e~00fb2~d6d2c4~ffc845~dc582a~41b6e6~1b365d~be84a3~5d3754~7da1c4~f2c75c~d0d3d4~007a78~000000](http://www.sussex.ac.uk/tel/resource/tel_website/accessiblecontrast/?q=FFFFFF~003b49~1d4289~94a596~e56db1~d3273e~00fb2~d6d2c4~ffc845~dc582a~41b6e6~1b365d~be84a3~5d3754~7da1c4~f2c75c~d0d3d4~007a78~000000)
- [38] CSUN – Universal Design Center. Web Accessibility Criteria - Color Contrast. Retrieved June 19, 2023 from <https://www.csun.edu/universal-design-center/web-accessibility-criteria-color-contrast#:~:text=Black%20text%20on%20a%20white%20background%20has%20the%20highest%20readability>
- [39] WAVE Web Accessibility Evaluation Tools. Retrieved June 19, 2023 from <https://wave.webaim.org/>
- [40] Stackoverflow. How do I setup simple firebase ajax request? Retrieved June 20, 2023 from <https://stackoverflow.com/questions/36165311/how-do-i-setup-simple-firebase-ajax-request>

- [41] Stackoverflow. Is there a way to send a user an invite to sign up using Firebase Auth. Retrieved June 20, 2023 from <https://stackoverflow.com/questions/65602925/is-there-a-way-to-send-a-user-an-invite-to-sign-up-using-firebase-auth>
- [42] Firebase. Send Confirmation Emails with Cloud Functions. Retrieved June 20, 2023 from <https://github.com/firebase/functions-samples/tree/Node-8/email-confirmation>
- [43] Mozilla. 2023. HTTP request methods. Retrieved June 20, 2023 from <https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>
- [44] Alex Mamo. 2021. How to reduce Firestore costs? Retrieved June 20, 2023 from <https://medium.com/firebase-tips-tricks/how-to-reduce-firebase-costs-8cb712473e83>
- [45] Google. Firestore. Usage and limits. Retrieved June 20, 2023 from <https://firebase.google.com/docs/firestore/quotas>
- [46] Google. Firestore. Understand reads and writes at scale. Retrieved June 20, 2023 from <https://firebase.google.com/docs/firestore/understand-reads-writes-scale>
- [47] Frank van Puffelen. 2023. StackOverflow. how to make firestore query cache shared between clients using Firestore Data Bundles & CDN. Retrieved June 20, 2023 from <https://stackoverflow.com/questions/76193461/how-to-make-firebase-query-cache-shared-between-clients-using-firebase-data-bu>
- [48] ParTech Media. 2020. REST API Development. Retrieved June 20, 2023 from <https://www.partech.nl/nl/publicaties/2020/07/9-trending-best-practices-for-rest-api-development#>
- [49] Geoapify API Pricing. Retrieved June 20, 2023 from <https://www.geoapify.com/pricing>
- [50] Google. Places API Usage and Billing. Retrieved June 20, 2023 from <https://developers.google.com/maps/documentation/places/web-service/usage-and-billing>
- [51] RapidAPI. Google Autocomplete APIs. Retrieved June 20, 2023 from <https://rapidapi.com/collection/google-autocomplete-api>
- [52] Martin Nally. 2017. Solving SEO problems with API design, part 2. Retrieved June 20, 2023 from <https://cloud.google.com/blog/products/api-management/solving-seo-problems-with-api-design-pt-2>
- [53] LocationIQ pricing. Retrieved June 20, 2023 from <https://locationiq.com/pricing>
- [54] Mapbox pricing. Retrieved June 20, 2023 from <https://www.mapbox.com/pricing>
- [55] MapQuest. Pricing Plans. Retrieved June 20, 2023 from <https://developer.mapquest.com/plans>
- [56] TomTom. API pricing. Retrieved June 20, 2023 from <https://developer.tomtom.com/store/maps-api>
- [57] Mapquest. Documentation. Retrieved June 20, 2023 from <https://developer.mapquest.com/documentation/>
- [58] Google. Firestore. Supported data types. Retrieved June 20, 2023 from <https://firebase.google.com/docs/firestore/manage-data/data-types>
- [59] Google. Firestore. Updating a Firestore document in a transaction. Retrieved June 20, 2023 from <https://cloud.google.com/firestore/docs/samples/firestore-transaction-document-update>
- [60] Google. Firestore. Get started with Cloud Firestore Security Rules. Retrieved June 20, 2023 from <https://firebase.google.com/docs/firestore/security/get-started>
- [61] Google. SendGrid Email API pricing. Retrieved June 20, 2023 from <https://console.cloud.google.com/marketplace/product/sendgrid-app/sendgrid-email>
- [62] Mailjet. Pricing. Retrieved June 20, 2023 from <https://www.mailjet.com/pricing/>
- [63] Mailgun. Pricing. Retrieved June 20, 2023 from <https://www.mailgun.com/pricing/>

- [64] Google Adsense. Retrieved June 20, 2023 from [https://support.google.com/adsense/answer/9902?hl=en&ref\\_topic=1319755&sjid=4304153593144887845-EU](https://support.google.com/adsense/answer/9902?hl=en&ref_topic=1319755&sjid=4304153593144887845-EU)
- [65] Stripe pricing. Retrieved June 20, 2023 from <https://stripe.com/en-lu/pricing>
- [66] Square pricing. Retrieved June 20, 2023 from <https://squareup.com/us/en/pricing>
- [67] Generating PDFs with Firestore extension. Retrieved June 20, 2023 from <https://extensions.dev/extensions/pdfplum/firestore-pdf-generator>
- [68] Testim. 2021. Node.js Unit Testing: Get Started Quickly With Examples. Retrieved June 20, 2023 from <https://www.testim.io/blog/node-js-unit-testing-get-started-quickly-with-examples/>
- [69] Raven Veal. 2022. How To Write Effective Usability Testing Questions: A Beginner's Guide. Retrieved June 20, 2023 from <https://careerfoundry.com/en/blog/ux-design/how-to-write-usability-testing-questions/>
- [70] Chandan Kumar. 2023. 14 Online Free Tools to Scan Website Security Vulnerabilities & Malware. Retrieved June 21, 2023 from <https://geekflare.com/online-scan-website-security-vulnerabilities/>
- [71] IntelliPaat. 2023. 10 Powerful Vulnerability Scanning Tools in 2023. Retrieved June 21, 2023 from <https://intellipaat.com/blog/vulnerability-scanning-tools/?US>
- [72] Robert Bateman. 2023. Legal Considerations Before Launching a New Website. Retrieved June 21, 2023 from <https://www.termsfeed.com/blog/legal-considerations-launching-new-website/>
- [73] SEO-Hacker. How to Choose the Right Domain Name. Retrieved June 21, 2023 from <https://seo-hacker.com/choose-right-domain-name/>
- [74] Backlinko. 2023. SEO Best Practices. Retrieved June 21, 2023 from <https://backlinko.com/hub/seo/best-practices>
- [75] PageSpeed Insights. 2023. Retrieved June 21, 2023 from [https://pagespeed.web.dev/?utm\\_source=psi&utm\\_medium=redirect](https://pagespeed.web.dev/?utm_source=psi&utm_medium=redirect)
- [76] Backlinko. 2023. Google Search Console: The Definitive Guide. Retrieved June 21, 2023 from <https://backlinko.com/google-search-console>
- [77] Brian Dean. 2023. 17 Advanced SEO Techniques for 2023. Retrieved June 21, 2023 from <https://backlinko.com/advanced-seo>
- [78] Brian Dean. 2022. How to Use Google Keyword Planner. Retrieved June 21, 2023 from <https://backlinko.com/google-keyword-planner>
- [79] Brian Dean. 2023. Backlinks. Retrieved June 21, 2023 from <https://backlinko.com/hub/seo/backlinks>
- [80] Google Search Central. Link best practices for Google . Retrieved June 21, 2023 from <https://developers.google.com/search/docs/crawling-indexing/links-crawlable#crawlable-links>
- [81] Michael Brig. Location icon in Inkscape. Retrieved June 22, 2023 from <https://www.youtube.com/watch?v=lBal6p9YAek&t=17s>
- [82] Davies Media Design. How to Draw a Heart Shape in Inkscape. Retrieved June 22, 2023 from <https://www.youtube.com/watch?v=7PByOz30YWI>
- [83] Crafty Stax. INKSCAPE: How To Cut an Object (or a Shape) With an Object. Retrieved June 22, 2023 from <https://www.youtube.com/watch?v=tdbQXUnLSVw>
- [84] Firebase. Add Firebase to your JavaScript project. Retrieved June 23, 2023 from <https://firebase.google.com/docs/web/setup>
- [85] Firebase. Get Started with Firebase Authentication on Websites. Retrieved June 23, 2023 from <https://firebase.google.com/docs/auth/web/start>

- [86] Firebase. Authentication State Persistence Retrieved June 23, 2023 from <https://firebase.google.com/docs/auth/web/auth-state-persistence>
- [87] Github. Can I provide a custom persistence implementation for auth in node? Retrieved June 24, 2023 from <https://github.com/firebase/firebase-js-sdk/issues/1874#issuecomment-1608395275>
- [88] StackOverflow. Harry. 2017. How to authenticate an user in firebase-admin in nodejs? Retrieved June 25, 2023 from <https://stackoverflow.com/questions/44899658/how-to-authenticate-an-user-in-firebase-admin-in-nodejs>
- [89] StackOverflow. HexaCrop. 2019. what is the authentication flow for a firebase and mean stack architecture? Retrieved June 25, 2023 from <https://stackoverflow.com/questions/56784748/what-is-the-authentication-flow-for-a-firebase-and-mean-stack-architecture>
- [90] Firebase. Verify ID Tokens. Retrieved June 26, 2023 from <https://firebase.google.com/docs/auth/admin/verify-id-tokens>
- [91] StackOverflow. Explorer. 2023. Is it good practice to use firebase-admin-sdk and firebase-ui + firebase-client-sdk together to handle server-side and client-side authentication? Retrieved June 26, 2023 from <https://stackoverflow.com/questions/75408384/is-it-good-practice-to-use-firebase-admin-sdk-and-firebase-ui-firebase-client>
- [92] StackOverflow. David Y. Stephenson. 2018. Handling Firebase ID tokens on the client side with vanilla JavaScript. Retrieved June 26, 2023 from <https://stackoverflow.com/questions/48884217/handling-firebase-id-tokens-on-the-client-side-with-vanilla-javascript>
- [93] Github. Firebase. 2023. Retrieved June 26, 2023 from <https://github.com/firebase/functions-samples/tree/main/Node-1st-gen/template-handlebars>
- [94] Firebase. Manage Session Cookies. Retrieved June 26, 2023 from <https://firebase.google.com/docs/auth/admin/manage-cookies>
- [95] Firebase. quickstart-nodejs. Retrieved June 26, 2023 from <https://github.com/firebase/quickstart-nodejs/blob/master/auth-sessions/app.js>
- [96] Firebase. Introduction to Firebase Local Emulator Suite. Retrieved June 26, 2023 from <https://firebase.google.com/docs/emulator-suite>
- [97] StackOverflow. Sagiv Ofek. 2017. firebase cloud function won't store cookie named other than "\_\_session". Retrieved June 27, 2023 from <https://stackoverflow.com/questions/44929653/firebase-cloud-function-wont-store-cookie-named-other-than-session>
- [98] Firebase. Using cookies. Retrieved June 27, 2023 from [https://firebase.google.com/docs/hosting/manage-cache#using\\_cookies](https://firebase.google.com/docs/hosting/manage-cache#using_cookies)
- [99] Bootstrap 5. Retrieved June 27, 2023 from <https://getbootstrap.com/docs/5.1>
- [100] mjade. 2020. Bootstrap V5 manually call a modal myModal.show() not working (vanilla javascript). Retrieved June 27, 2023 from <https://stackoverflow.com/questions/62827002/bootstrap-v5-manually-call-a-modal-my-modal-show-not-working-vanilla-javascript>
- [101] Bootstrap 5. Login form. Retrieved June 27, 2023 from <https://mdbootstrap.com/docs/standard/extended/login/>
- [102] StackOverflow. Skizit. 2011. How to set custom validation messages for HTML forms? Retrieved June 27, 2023 from <https://stackoverflow.com/questions/5272433/how-to-set-custom-validation-messages-for-html-forms>
- [103] StackOverflow. V.Aleksanyan. 2017. How I can sanitize my input values in node js? Retrieved June 27, 2023 from <https://stackoverflow.com/questions/46718772/how-i-can-sanitize-my-input-values-in-node-js>

- [104] express-validator. Retrieved June 27, 2023 from <https://express-validator.github.io/docs/>
- [105] Yoast. 2021. Preventing your site from being indexed, the right way. Retrieved June 28, 2023 from <https://yoast.com/prevent-site-being-indexed/>
- [106] Firebase. Firebase Security Rules. Retrieved June 28, 2023 from <https://firebase.google.com/docs/rules>
- [107] Firebase. Closed access. Retrieved June 28, 2023 from [https://firebase.google.com/docs/rules/insecure-rules#closed\\_access](https://firebase.google.com/docs/rules/insecure-rules#closed_access)
- [108] Jash Unadkat. 2023. Top JavaScript Testing Frameworks. Retrieved June 29, 2023 from <https://www.browserstack.com/guide/top-javascript-testing-frameworks>
- [109] Amit Kumar Gupta. 2018. Mock HTTP request and response with request payload. Retrieved June 29, 2023 from <https://stackoverflow.com/questions/49491091/mock-http-request-and-response-with-request-payload>
- [110] Node.js v18.16.1 documentation. Test runner. Retrieved June 29, 2023 from <https://nodejs.org/docs/latest-v18.x/api/test.html#test-runner>
- [111] Gleb Bahmutov. 2023. Trying Node.js Test Runner <https://glebbahmutov.com/blog/trying-node-test-runner/>
- [112] Github. node-mocks-http. Retrieved June 29, 2023 from <https://github.com/eugef/node-mocks-http>
- [113] StackOverflow. DAVID TEC . 2017. Run a web socket on Cloud Functions for Firebase?. Retrieved June 30, 2023 from <https://stackoverflow.com/questions/45741284/run-a-web-socket-on-cloud-functions-for-firebase>
- [114] Firebase. Firebase Realtime Database. Retrieved June 30, 2023 from <https://firebase.google.com/docs/database>
- [115] Firebase. Understand Realtime Database Billing. Retrieved June 30, 2023 from <https://firebase.google.com/docs/database/usage/billing>
- [116] Firebase. Understand the real-time query system. Retrieved June 30, 2023 from [https://firebase.google.com/docs/firestore/real-time\\_queries\\_at\\_scale#understand\\_the\\_real-time\\_query\\_system](https://firebase.google.com/docs/firestore/real-time_queries_at_scale#understand_the_real-time_query_system)
- [117] Timonwa Akintokun. 2023. How to Build a Real-time Chat App with ReactJS and Firebase. Retrieved June 30, 2023 from <https://www.freecodecamp.org/news/building-a-real-time-chat-app-with-reactjs-and-firebase/>
- [118] John Rodkey. 2020. Directly connect your frontend and backend using Firebase. Retrieved June 30, 2023 from <https://blog.logrocket.com/directly-connect-your-frontend-and-backend-using-firebase/>
- [119] AlterClass. 2021. Build a REALTIME CHAT APP in 5 MIN with React and Firebase. Retrieved June 30, 2023 from <https://www.youtube.com/watch?v=2-LISBTczQE> (min 6:40).
- [120] Firebase. Get realtime updates with Cloud Firestore. Retrieved June 30, 2023 from <https://firebase.google.com/docs/firestore/query-data/listen>
- [121] Firebase Hosting. Retrieved July 16, 2023 from <https://firebase.google.com/docs/hosting>
- [122] Github. Susanna Ventafridda. 2023. Final project <https://github.com/SusyVenta/UniFinalProject>
- [123] Hostedscan.com. Retrieved September 10<sup>th</sup> 2023 from <https://hostedscan.com/>
- [124] StackOverflow. Set cookies for cross origin requests. Abdullah Oladipo. 2020. Retrieved September 10<sup>th</sup> 2023 from <https://stackoverflow.com/questions/46288437/set-cookies-for-cross-origin-requests>
- [125] Helmet. Retrieved September 10<sup>th</sup> 2023 from [https://www.npmjs.com/package/helmet.](https://www.npmjs.com/package/helmet)

- [126] StackOverflow. How to set X-Frame-Options in express.js node.js. Edwin O. 2019. Retrieved September 10<sup>th</sup> 2023 from <https://stackoverflow.com/questions/46998540/how-to-set-x-frame-options-in-express-js-node-js> .
- [128] StackOverflow. Firebase Callbacks - what's the underlying trigger? Zo72. 2013. Retrieved September 16<sup>th</sup> 2023 from <https://stackoverflow.com/questions/18638917/firebase-callbacks-whats-the-underlying-trigger>
- [129] GroupTripper. Security rules. Susanna Ventafridda. 2023. Retrieved September 16<sup>th</sup> 2023 from <https://github.com/SusyVenta/UniFinalProject/blob/main/firestore.rules>
- [130] GroupTripper. Database interactions – generic operations. Susanna Ventafridda. 2023. Retrieved September 16<sup>th</sup> 2023 from <https://github.com/SusyVenta/UniFinalProject/blob/main/functions/db/db.js>
- [131] GroupTripper. Database interactions – specific queries. Susanna Ventafridda. 2023. Retrieved September 16<sup>th</sup> 2023 from <https://github.com/SusyVenta/UniFinalProject/tree/main/functions/db/queries>
- [132] GroupTripper. Example of onSnapshot listener. Susanna Ventafridda. 2023. Retrieved September 16<sup>th</sup> 2023 from <https://github.com/SusyVenta/UniFinalProject/blob/main/public/js/tripPolls.js#L751>
- [133] Firebase – transactions. Google. Susanna Ventafridda. 2023. Retrieved September 16<sup>th</sup> 2023 from <https://firebase.google.com/docs/firestore/manage-data/transactions>
- [134] Github – Croppie. 2023. Retrieved September 16<sup>th</sup> 2023 from <https://foliotek.github.io/Croppie/>
- [135] Codepen – Croppie example. 2023. Retrieved September 16<sup>th</sup> 2023 from <https://codepen.io/rsales/pen/XyWORN>
- [136] Node.js library – Croppie. 2023. Retrieved September 16<sup>th</sup> 2023 from <https://www.npmjs.com/package/croppie>
- [137] GroupTripper initialization of Croppie from HTML. Susanna Ventafridda. 2023. Retrieved September 16<sup>th</sup> 2023 from <https://github.com/SusyVenta/UniFinalProject/blob/main/functions/views/profile.ejs#L259>
- [138] GroupTripper initialization of Croppie in JS. Susanna Ventafridda. 2023. Retrieved September 16<sup>th</sup> 2023 from <https://github.com/SusyVenta/UniFinalProject/blob/main/public/js/profile.js#L45>
- [139] Tom-select. 2023. Retrieved September 16<sup>th</sup> 2023 from <https://tom-select.js.org/docs/api/>
- [140] StackOverflow. Adding bootstrap 5 search bar dropdown. Tester321. 2021. Retrieved September 16<sup>th</sup> 2023 from <https://stackoverflow.com/questions/69530889/adding-bootstrap-5-search-bar-dropdown>
- [141] GroupTripper. Tom-select used to add friends in trip poll. Susanna Ventafridda. 2023. Retrieved September 16<sup>th</sup> 2023 from <https://github.com/SusyVenta/UniFinalProject/blob/main/public/js/tripPolls.js#L34>
- [142] GroupTripper entry point. Susanna Ventafridda. 2023. Retrieved September 16<sup>th</sup> 2023 from <https://github.com/SusyVenta/UniFinalProject/blob/main/functions/index.js>
- [143] GroupTripper routers. Susanna Ventafridda. 2023. Retrieved September 16<sup>th</sup> 2023 from <https://github.com/SusyVenta/UniFinalProject/tree/main/functions/routers>
- [144] GroupTripper – settings router. Susanna Ventafridda. 2023. Retrieved September 16<sup>th</sup> 2023 from <https://github.com/SusyVenta/UniFinalProject/blob/main/functions/routers/settings.js>
- [145] GroupTripper – Ajax calls. Susanna Ventafridda. 2023. Retrieved September 16<sup>th</sup> 2023 from <https://github.com/SusyVenta/UniFinalProject/blob/main/public/js/tripPolls.js#L183>

## Appendix

### Appendix A – Market research

#### A.1 – Competitors analysis summary

Relevance	Tool	Pros	Cons	GroupTripper opportunities
1	<p><a href="https://www.troupe.com/">https://www.troupe.com/</a></p> <p>PLAN GROUP TRIPS WITHOUT THE HEADACHE.</p> <p>Say goodbye to long email threads and group texts. Troupe is your one stop shop to plan a group trip everyone will love.</p> <p>Invite friends to vote on important details like when to go, where to stay, and what to do. Now that's group travel untangled.</p>	<ul style="list-style-type: none"> <li>▪ Specifically built to create a trip plan as a group.</li> <li>▪ The organizer sets a time window within which collaborators can propose dates, destination, stay, and activities. The organizer can then close the time window to propose and open a time window to allow everyone to vote on the proposed options. Once this time window is closed, the option with highest votes is chosen. The organizer can then create an itinerary that includes only the chosen options.</li> <li>▪ Integrates with Expedia API to search for accommodation in app.</li> <li>▪ Lets you explore features before signing up</li> </ul>	<ul style="list-style-type: none"> <li>▪ No section to propose options on transportation</li> <li>▪ Does not allow multi-destination trips</li> <li>▪ For each category, it only lets choose one item (one set of dates, one destination, one stay)</li> <li>▪ Requires to manually create an itinerary instead of automatically building it from the choices made.</li> <li>▪ Only iOS App, no Android</li> <li>▪ Does not allow to vote on different trip proposals as a whole. Forces to first choose dates, then location etc. However, choosing dates in most cases depends on price for flights and hotels on the dates. Need to have visibility on</li> </ul>	<ul style="list-style-type: none"> <li>▪ Expands on the features available but allow the group to propose full plans and vote on the whole</li> <li>▪ Add transportation options and multi-destination trips</li> <li>▪ Automatically save the chosen itinerary</li> <li>▪ Ensure the site can be found with common search terms used by Wanderlog.com “trip planner app”, “travel planner app”</li> <li>▪ [After MVP] Integrate with TripAdvisor API to propose activities by location (potential revenue strategy via applying to affiliate links program) and / or <a href="#">Google Maps API (pricing)</a> to search for places nearby or free search with <a href="#">OpenStreetMap or alternatives</a>.</li> <li>▪ Will make it available in different languages: Italian,</li> </ul>

			<ul style="list-style-type: none"> <li>▪ the whole thing to decide.</li> <li>▪ Does not propose activities by location</li> <li>▪ <a href="#"><u>SEO seems unoptimized.</u></a> People don't find the tool with keyword "trip planner" or "travel planner" as opposed to <a href="#"><u>Wanderlog.com</u></a>.</li> <li>▪ Only advertises on YouTube and Pinterest</li> <li>▪ Only available in English (hence used only in English-speaking countries)</li> </ul>	Spanish, French to begin with.
2	<p><a href="https://wanderlog.com/">https://wanderlog.com/</a></p> <p>You'll never travel without our trip planner again Build, organize, and map your itineraries in a free travel app designed for vacations &amp; road trips</p>	<ul style="list-style-type: none"> <li>▪ Good features to save plan itineraries, hotels, transportations, and activities.</li> <li>▪ Activities recommendations</li> <li>▪ Scratch-map feature allows to save visited places with a pin</li> <li>▪ In-app search of hotels</li> <li>▪ Allows forwarding emails with bookings, parses content and adds to plan</li> <li>▪ Freemium plan. Pro features: adding attachments, rental car deals, dark mode</li> <li>▪ <a href="#"><u>Good SEO:</u></a> people find the site by googling "trip planner", "itinerary planner", "travel planner"</li> <li>▪ Good advertisement on Facebook, Reddit,</li> </ul>	<ul style="list-style-type: none"> <li>▪ Assumes that you have already decided on a destination and dates</li> <li>▪ Does not allow friends to vote on options</li> <li>▪ Scratch-map feature doesn't allow comparison with different friends. Would be nice to have pop up notifications recommending common destinations that friends want to visit.</li> <li>▪ No clear way to add friends from profile</li> <li>▪ Doesn't allow to assign status to</li> </ul>	<ul style="list-style-type: none"> <li>▪ Allows to choose a destination together by comparing different options as a group</li> <li>▪ [After MVP] Has scratch-map feature that allows to compare visited countries with friends</li> <li>▪ Allows to enter visited places and places you want to visit. Spots common desired destinations and proposes trips with notifications.</li> <li>▪ User friendly way to add friends</li> <li>▪ Have status field for trips and allow easy search (hide completed trips)</li> </ul>

		<ul style="list-style-type: none"> <li>Youtube, Instagram. Uses Google Adsense</li> <li>▪ Can add custom or default todo lists to each step of itinerary</li> </ul>	<ul style="list-style-type: none"> <li>trips and filtering by status.</li> <li>▪ Doesn't add completed trips to map automatically.</li> <li>▪ Requires sign-up before exploring features</li> <li>▪ Only available in English</li> </ul>	<ul style="list-style-type: none"> <li>▪ [After MVP] Will also use a freemium model</li> <li>▪ [After MVP] Will also advertise with Adsense and on Facebook, Insta, Youtube.</li> <li>▪ Will allow to add custom or default todo lists</li> <li>▪ Will allows users to explore features without signing up</li> </ul>
3	<p><a href="https://www.pilotplans.com/">https://www.pilotplans.com/</a></p> <p>The Social Trip Planning Tool To make the most of every trip</p>	<ul style="list-style-type: none"> <li>▪ Intuitive way to create a multi-destination travel plan, including accommodation, transportation, and activities</li> <li>▪ Adds chosen destination to a map</li> <li>▪ Intuitive to add friends</li> <li>▪ Allows to add notes and files</li> <li>▪ In-app chat to discuss with friends added to the trip</li> <li>▪ Advertises on LinkedIn, YouTube, Facebook, Google Adsense</li> </ul>	<ul style="list-style-type: none"> <li>▪ Does not recommend activities</li> <li>▪ Not though to collaborate and vote on a plan</li> <li>▪ <u>Not optimized SEO</u> – people don't reach the site via meaningful keywords</li> <li>▪ Only available in English</li> <li>▪ No Android App</li> </ul>	<ul style="list-style-type: none"> <li>▪ Will leverage the intuitive design to add items to the trip plan</li> <li>▪ [After MVP] Will integrate the map view – automatically adding location on a map once the user adds them to the list.</li> <li>▪ Will implement integrated chat that allows discussing each trip with participants.</li> </ul>
4	<p><a href="https://planify.io/">https://planify.io/</a></p> <p>“Plan, share, engage”</p> <p>Planify saves you time and removes the complexity of gathering information from multiple partners. Create an up-to-date, changeable and attractive</p>	<ul style="list-style-type: none"> <li>▪ Allows to create a plan by entering start date and name of plan. Then allows to add activities with detailed description such as start and end datetime, type of activity (transportation, hotel check-in etc.).</li> <li>▪ Allows to download plan as a PDF.</li> <li>▪ Different languages supported</li> </ul>	<ul style="list-style-type: none"> <li>▪ No Group collaboration features.</li> <li>▪ Unintuitive UI to add activities and reorganize them (can't shuffle them once created)</li> <li>▪ Unfortunately, <u>no web traffic statistics are available</u>.</li> <li>▪ Need to commit to a trial before</li> </ul>	<ul style="list-style-type: none"> <li>▪ Will integrate activities details like start and end time, address, URL, adding documents</li> <li>▪ Will allow to download plans as PDF (maybe premium feature)</li> </ul>

	plan to share with your participants.	<ul style="list-style-type: none"> <li>▪ <a href="#">3 different payment plans</a> with 2 weeks trial</li> <li>▪ Has Android app</li> </ul>	being able to explore features	
5	<p><a href="https://www.planharmony.com/">https://www.planharmony.com/</a></p> <p>Bring harmony to planning travel with friends</p>	<ul style="list-style-type: none"> <li>▪ Free</li> <li>▪ Creates a calendar view that can be exported to Google calendar</li> <li>▪ Allows to browse activities by location (Trip Advisor API integration)</li> <li>▪ Trip budget and bill splitting functionality</li> <li>▪ Todo lists</li> <li>▪ Voting</li> <li>▪ Map view</li> </ul>	<ul style="list-style-type: none"> <li>▪ No mobile version</li> <li>▪ Voting functionality only allows to add positive votes. No action is then taken with votes.</li> <li>▪ Feature proposing activities by location doesn't work (in beta)</li> </ul>	<ul style="list-style-type: none"> <li>▪ Like the fact that there is one way to add events. One form in which based on the selected event category, one can fill in different fields.</li> </ul>
6	<p><a href="https://mitravelapp.com/">https://mitravelapp.com/</a></p> <p>MiTravel is a virtual planning board for travel enthusiasts and avid planners to collaboratively plan and visualize their best trips together.</p>	<ul style="list-style-type: none"> <li>▪ Create polls to make decisions quickly</li> </ul>	<ul style="list-style-type: none"> <li>▪ Limited features</li> <li>▪ Kanban board looks chaotic and does not finalize plan. More for brainstorming but format does not look like a tidy plan.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Will have custom poll functionality</li> </ul>
7	<p><a href="https://travelmobili.com/">https://travelmobili.com/</a></p> <p>Providing travelers with an organizational tool that eliminates the headaches of gathering everyone's information.</p> <p>Mobili takes the hassle out of travel planning for individuals and groups! Track, manage and organize your trip</p>	<ul style="list-style-type: none"> <li>▪ Allows to enter individual travel details per each group member (if travelers start from different places)</li> <li>▪ Allows to track group members position on the map at the time the trip is active</li> <li>▪ Spend tracking functionality</li> </ul>	<ul style="list-style-type: none"> <li>▪ No website, only apps</li> <li>▪ No voting capabilities</li> </ul>	<ul style="list-style-type: none"> <li>▪ Will include functionality to specify what group members participate to which activities (allow some activities to be shared only by a subset of group)</li> <li>▪ Will have spend tracking functionality</li> </ul>

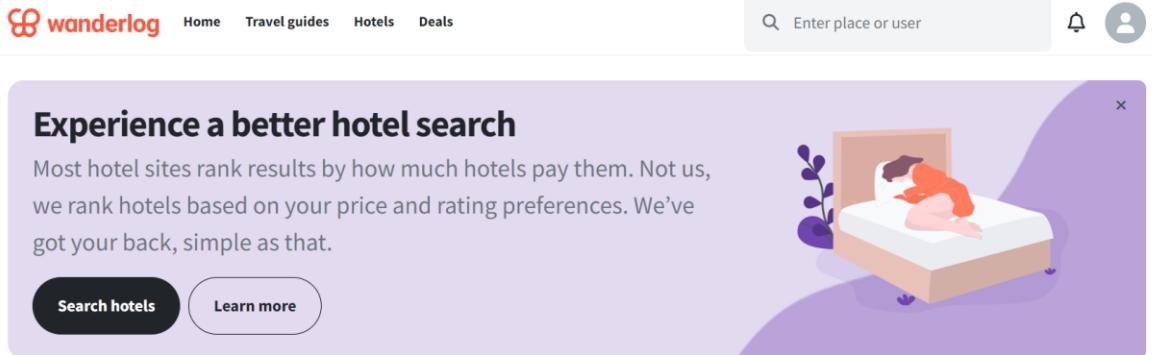
	<p>details, accommodation and travel itinerary in one handy place. Traveling with a group? Share and compare your details side by side in real-time with everybody in the group, use the in-app messenger or see their location in real time! No more stress, no more fuss. Focus on your bags – we'll handle the rest!</p>			
8	<a href="https://www.tripit.com/web/free">https://www.tripit.com/web/free</a>	<ul style="list-style-type: none"> <li>▪ Allows to enter dates and location of a trip and forward booking confirmations to an email address to create a trip plan that can be exported to calendar and shared</li> <li>▪ Freemium plan. Pro plan allows to track plane seats, notifies on plane ticket fares, check-in reminder, shares travel plan with friends, provides country specific travel information, allows to add documents, photos etc.</li> <li>▪ Multiple language support</li> </ul>	<ul style="list-style-type: none"> <li>▪ No group planning collaboration features</li> </ul>	<ul style="list-style-type: none"> <li>▪ [After MVP] Will consider adding this feature, which is also present on Wanderlog.com</li> </ul>
9	<a href="https://hoku.travel/">https://hoku.travel/</a> Hoku   The best trip planner to organize travel with friends		<ul style="list-style-type: none"> <li>▪ No website to try out functionality</li> <li>▪ Can't <a href="#">download the app</a> on <a href="#">PlayStore</a></li> </ul>	<ul style="list-style-type: none"> <li>▪ Will have both a website and [After MVP] an app for Android and iOS</li> </ul>

	Our trip planner helps friends organize all travel-related plans and explore activities around the world, so you can spend less time planning your vacations.			
10	<a href="https://github.com/hucki/grouptripper">https://github.com/hucki/grouptripper</a>	<ul style="list-style-type: none"> <li>▪ Open source project found online that has functionality to add locations to a map with itinerary</li> </ul>	<ul style="list-style-type: none"> <li>▪ No hosted service. Just a thesis project</li> </ul>	<ul style="list-style-type: none"> <li>▪ Will assess if I can leverage any of their existing functionality, taking inspiration on displaying destinations on a map</li> </ul>

## A.2 – Visual collection of competitors’ solutions

### Wanderlog

#### 1) Home



The banner features the Wanderlog logo and navigation links for Home, Travel guides, Hotels, and Deals. A search bar with placeholder text "Enter place or user" is positioned at the top right, along with a bell icon and a user profile icon. The main headline reads "Experience a better hotel search" and emphasizes that they rank hotels based on price and rating preferences, not by how much they pay. Below the headline are two buttons: "Search hotels" and "Learn more". To the right of the text is a purple-toned illustration of a person sleeping peacefully in a bed.

#### Recently viewed and upcoming

+ Plan new trip



Trip to Strasbourg

5 days • May 1 – Jun 30 • 5 places

#### Hotels ranked by price and rating preferences

Experience a better hotel search. We rank hotels based on your price and rating preferences, not by how much they pay us. We also show you the total cost upfront, among other benefits. [Learn more](#).

Where  
City name

When  
5/23 – 5/24

Rooms, guests  
1 2

Search for bookings



The search interface includes a map of Europe with highlighted locations. At the top left, there are filters for "COUNTRIES" (set to 2), "CITIES & REGIONS" (set to 2), and a "Novice Explorer" badge. On the right, there is a "Search for bookings" button and a "Update map" button with a speech bubble icon.



**Your trips**

Trip to Strasbourg

S • May 1 – Jun 30 • 5 places

[+ Plan new trip](#)

**Your guides**

You don't have any guides yet. [Create a new guide](#).

[+ Create new guide](#)

## Explore

### Popular destinations



**Siem Reap Guide (Angkor Temples)**

I live in Bangkok and travel Southeast Asia extensively. This is a general guide covering the key...

Gary W

0 56

Allan Watt

6 days in Iceland

Our visit to San Francisco was part of a road trip we enjoyed on the Pacific West Coast of the USA startin...

0 403

Naomi Wuarbanaran

7 350

**San Francisco, USA**

Our visit to San Francisco was part of a road trip we enjoyed on the Pacific West Coast of the USA startin...

Allan Watt

6 days in Iceland

We were in Iceland in February 2023 for 5 days. The tips and itenerary are applicable for every time of t...



Made with ❤ in SF & more  
© 2023 Travelchime Inc.

### Wanderlog

Blog

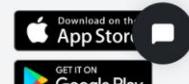
Report security issue

### Guides and resources

Trip planners by destination

Road trips by destination

### Get the app



2) Plan new trip

## Plan a new trip

**Where to?** e.g. Paris, Hawaii, Japan

Choose a destination to start planning

Dates (optional)

Start date

End date

+ Invite tripmates

Friends ▾

Friends

Public

Private

Start planning

Or write a new guide

### 3) Email request to join plan

l.it invited you to edit "Trip to Rome" on Wanderlog

✉ Posta in arrivo x

 **Wanderlog** <no-reply@wanderlog.com> [Annulla iscrizione](#)  
a me ▾

11:10 (1 minuto fa)  



**Hey there! [Sergio Vassalli](#) invited you to join the "Trip to Rome" trip plan on Wanderlog:**

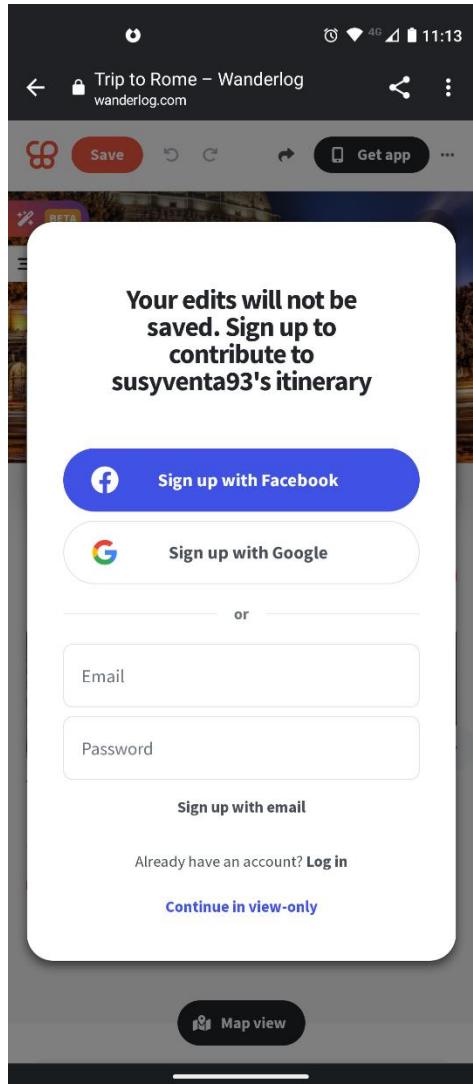
I'm planning my trip on Wanderlog and wanted to share it with you!

[Go to the trip](#)

Wanderlog lets you map and plan your trips with travel recommendations from your friends and from around the web. [Learn more](#).

If you don't know the inviter, please just ignore and delete this email; sorry for the bother!

#### 4) Invitation to sign up



## 5) trip view

**Trip Plan Assistant BETA**

**Overview**

- Explore
- Notes
- Places to visit
- Untitled

**Itinerary**

Thu 6/1	Explore
Fri 6/2	
Sat 6/3	
Sun 6/4	
Mon 6/5	
Tue 6/6	Top places for Rome Most often-seen on the web
Wed 6/7	
Thu 6/8	
Fri 6/9	
Sat 6/10	
Sun 6/11	
Mon 6/12	
Tue 6/13	
Wed 6/14	
Thu 6/15	
Fri 6/16	
Sat 6/17	
Sun 6/18	
Mon 6/19	
Tue 6/20	
Wed 6/21	

**Explore**

Browse all

Top places for Rome  
Most often-seen on the web

Best restaurants in Rome  
Most often-seen on the web

Wanderlog

**Reservations and attachments**

Flights Lodging Rental cars Attachment Other

**Budgeting**

\$0.00 View details

**Notes**

Write or paste anything here: how to get around, tips and tricks

**Places to visit**

Add a place

Export PRO

Add some places

Try typing Vatican Museums into one of these fields on the left:

Add a place

Or explore Top places for Rome

## 6) discovering activities

[← Explore](#)

Rome

### Categories

See all

- Restaurants
- Attractions
- Cafes
- Photo spots
- Cheap eats
- Breakfast and brunch
- Other eats

### Need a place to stay?

Where: Rome

When: 6/1 - 7/31

Rooms, guests: 1, 2

[Search for bookings](#)

### Top searches

- Pasta
- Hostels
- Carbonara
- Tiramisu

### Featured guides

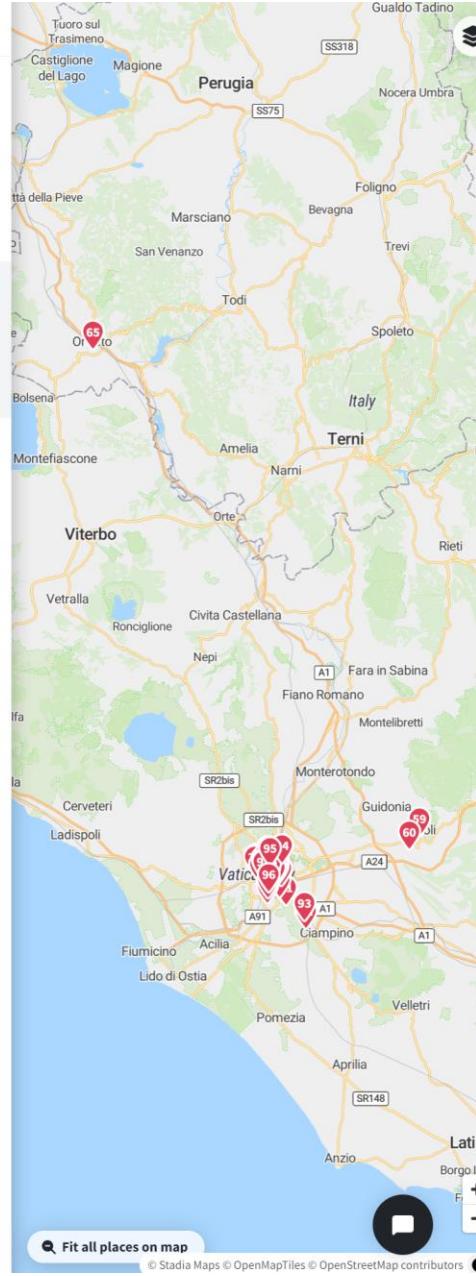
See all

- Brandon Jackson's History & Architecture in Rome Popular guide by a Wanderlog community member
- Purple & Yellow Backpackers's Lovely we... Popular guide by a Wanderlog community member
- Francesca Brooking's How to Spend 4 Days in Rome,... Popular guide by a Wanderlog community member

Brandon Jackson

### Multi-city stops

- Rome to ...
- Rome to ...
- Rome to ...
- Rome to ... Castel Nuovo del ...



7) Attaching flights info by sending email

en-seen on the web

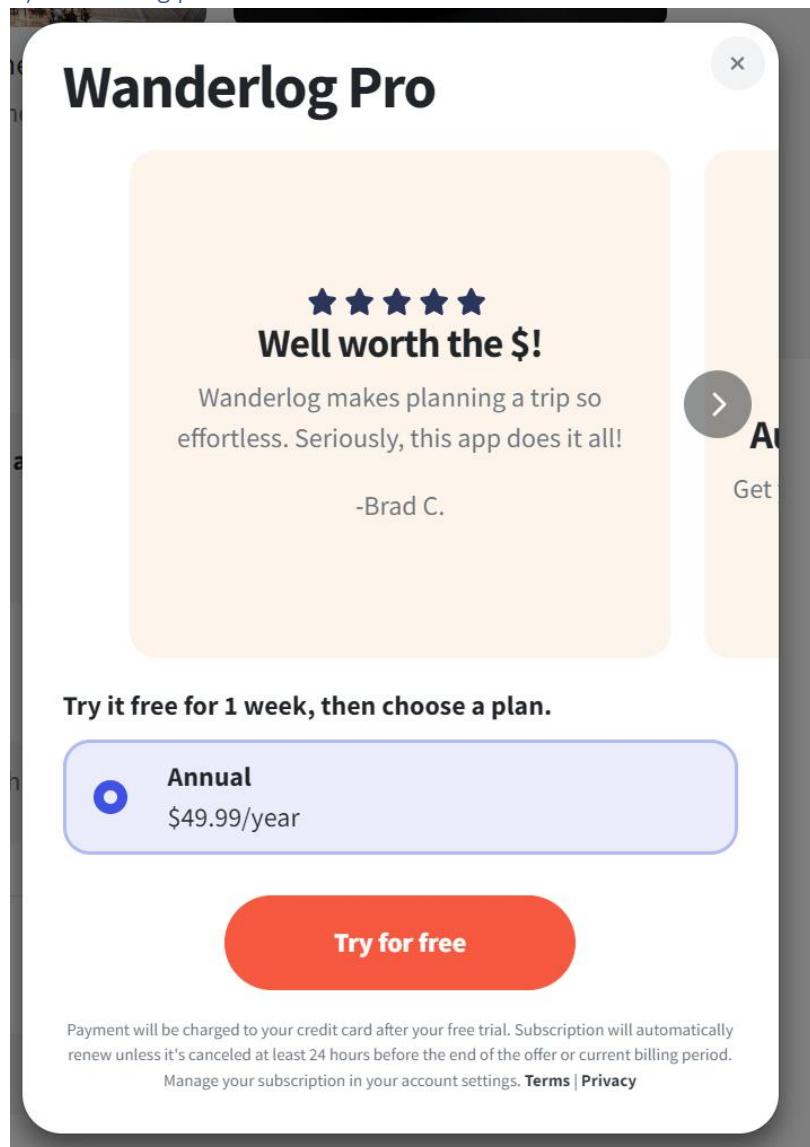
Most often-seen on the web

nderlog

Wanderlog



8) Wanderlog pro



The image shows a promotional screen for the Wanderlog Pro app. At the top, it says "Wanderlog Pro". Below that is a testimonial section with five blue stars and the text "Well worth the \$!". It includes a quote from Brad C.: "Wanderlog makes planning a trip so effortless. Seriously, this app does it all!" To the right, there's a "Get it now" button with a "Get" button and a "Get it on the App Store" link. Below the testimonial, it says "Try it free for 1 week, then choose a plan." A blue button highlights the "Annual" plan at \$49.99/year. At the bottom, there's a "Try for free" button and a note about automatic renewal.

**Wanderlog Pro**

★★★★★

**Well worth the \$!**

Wanderlog makes planning a trip so effortless. Seriously, this app does it all!

-Brad C.

Get it now

Get it on the App Store

Try it free for 1 week, then choose a plan.

**Annual**  
\$49.99/year

Try for free

Payment will be charged to your credit card after your free trial. Subscription will automatically renew unless it's canceled at least 24 hours before the end of the offer or current billing period.  
Manage your subscription in your account settings. [Terms](#) | [Privacy](#)

**Wanderlog Pro**

**Unlock our premium features:**

- ✓ Offline access
- ✓ Flight and car rental deals
- ✓ Unlimited attachments
- ✓ Export to Google Maps
- ✓ Optimize route
- ✓ Auto Gmail scanning
- ✓ Dark mode in app

[See more](#)

Try it free for 1 week, then choose a plan.

**Annual**  
\$49.99/year

[Try for free](#)

Payment will be charged to your credit card after your free trial. Subscription will automatically renew unless it's canceled at least 24 hours before the end of the offer or current billing period.  
Manage your subscription in your account settings. [Terms](#) | [Privacy](#)

## 9) Adding accommodation details

Add rental cars

**PRO ONLY DEAL**

Get exclusive discounts from our partners

**Hertz** UP TO 20% OFF   **AVIS** UP TO 35% OFF   **Budget** UP TO 35% OFF

or

Already booked your accommodations?

Forward email

Sync with Gmail **PRO**

Enter a city/airport

## 10) Adding items to trip plan

### Places to visit

...  
Add a place  
Add a note  
Add a checklist

Add a place

Recommended places

Colosseum + Trevi Fountain + >

11) budgeting and expenses feature

# Budgeting

+ Add expense

\$0.00

Set budget    Debt summary

View breakdown    Add tripmate    Settings

## Expenses

You haven't added any expenses yet.

Sort: Date ▾

**Split expenses with friends**

Add an expense, tell us who paid and whom to split it between – Wanderlog will track how much everyone owes automatically.

Got it

### Add expense

\$ 0

Select item >

Paid by: You ▾ Split: Don't split ▾

Date: Optional ▾

Save

### Select item

Select from your trip plan  
You don't have any places added yet

Or select from a category

Flights	Lodging	Car rental	Public transit
Food	Drinks	Sightseeing	Activities
Shopping	Gas	Groceries	Other

## 12) Profile

wanderlog

Home Travel guides Hotels Deals

Enter place or user

Novice Explorer

2 COUNTRIES 2 CITIES & REGIONS

Travel leaderboard

Update map

Add profile picture

Add your name  
@susyventa93

0 FOLLOWERS 0 FOLLOWING

Edit Share

In 23 days

Trip to Rome

S Jun 1 – Jul 31 0 places

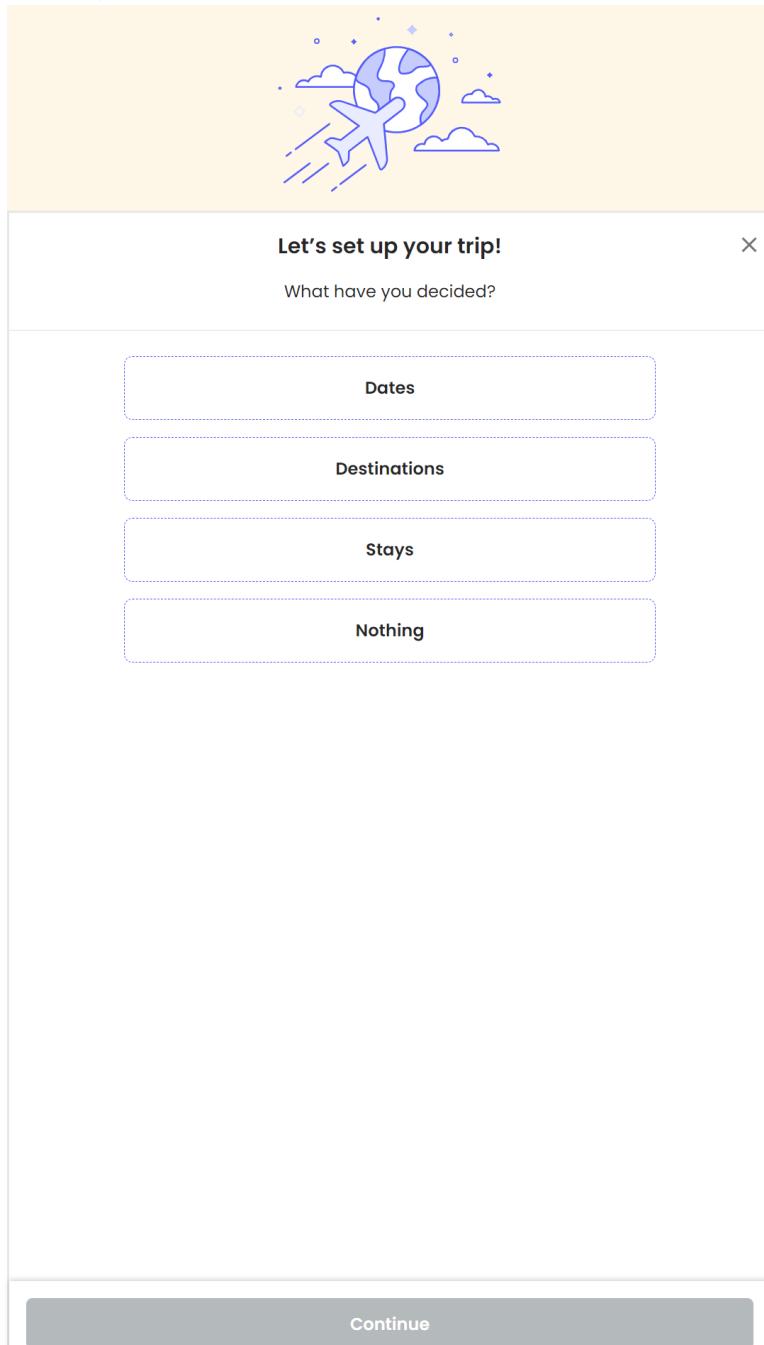
Trip plans Guides

Trip to Strasbourg

S May 1 – Jun 30 5 places

*Troupe*

1) Adding plan



2) travel details



## Finishing touches

X

Let travelers know what your trip is all about.

### Trip name

summer holidays at the seaside

30/100

Paragraph

v

B

U

Q

A

v

A

v

I

S

:=

!:=

:=

■

v

v

v

■

v

⋮

want to go to the seaside this summer

### Estimated travelers (optional)

Not including you, how many travelers do you think will join this trip?

4

### Add cover photo

Cover photos help your group get excited about where you're headed.

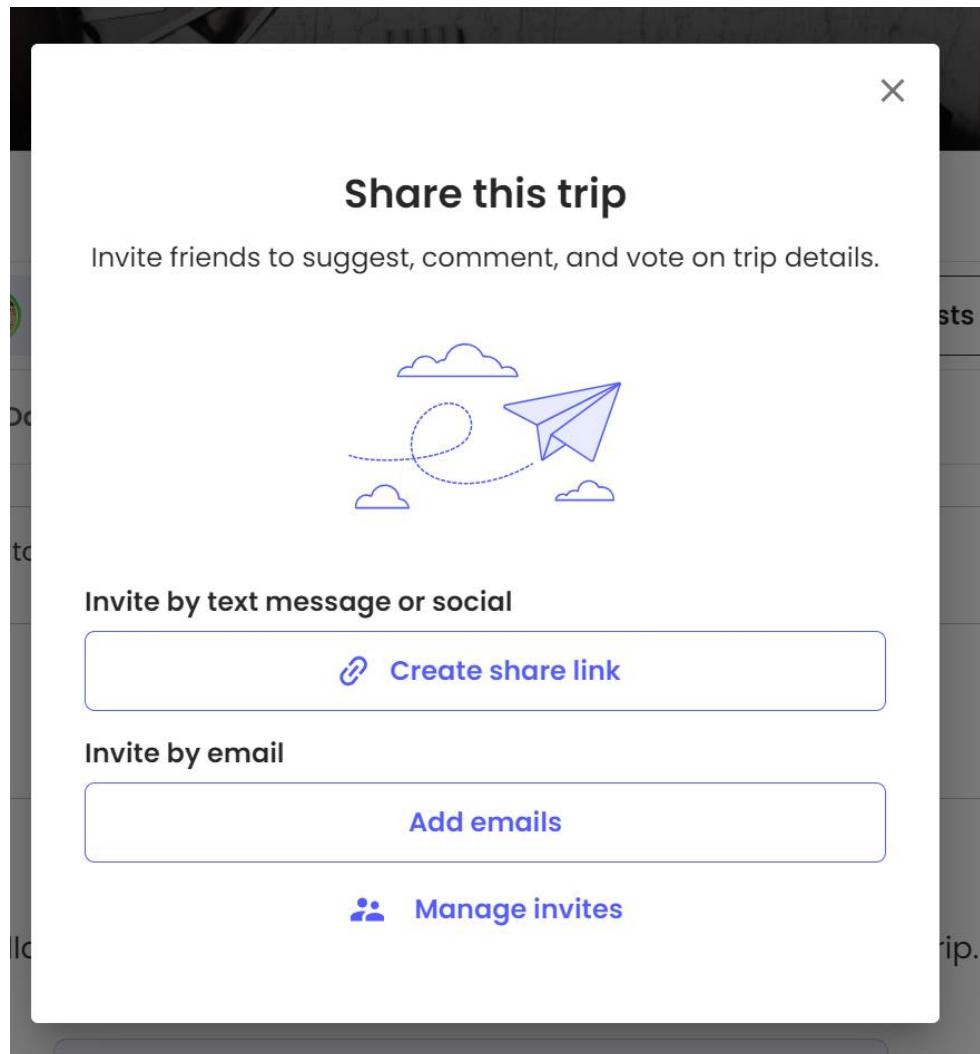


Click the camera icon to add a custom photo

Back

Submit

3) sharing trip



4) confirmation email

You've been invited to join "summer holidays at the seaside" ▶ Posta in arrivo ×



 SUSANNA VENTA (via Troupe) <concierge@troupe.com>  
a me ▾

12:30 (0 minuti fa)



# TROUPE



SUSANNA V. invited you to join a trip on Troupe

summer holidays at the seaside



test

Respond and let SUSANNA know if you can make the trip:

 Going

 Interested

## How does Troupe work?

Troupe is a group travel app that helps travelers suggest, vote, and decide on the most important parts of their trip.



**RSVP** to the trip



**Poll** your group



**Suggest** dates, destinations, stays, and activities



**Vote** on your favorites



**Build** your itinerary



**Vote** on your favorites



**Build** your itinerary

Want to see what's been planned so far?

[Explore and RSVP](#)

Safe travels,

**The Troupe Team**



---

Sent from **Troupe HQ**

12 E. 49th Street, New York, NY 10017

[5\) plan view](#)



### Trip Status

Collaborate with your group and keep track of decisions for this trip.

Dates

Add suggestions

Pick a date that works for the group.

Destinations

Add suggestions

Select the best destination for your trip.

Stays

Search and add stays

Find the perfect place to stay.

### The Who

Here's who's joined this trip so far.



SUSANNA ...  
*Going*

+  
Invite

Manage

6) defining dates

## Suggest dates

X

Suggest start and end dates for your group's upcoming trip.

### Suggestion 1

Start date  End date  

### Suggestion 2

Start date  End date  

[+ Add more dates](#)

 Save

## 7) defining dates - voting

summer holidays at the seaside

SUSANNA Planner Invite Broadcasts

Overview Polls Dates Destinations Stays Activities Itinerary Notes

**Dates**

Suggest new dates using the "+ Add dates" button below.

1 Suggest dates — 2 Vote — 3 Finalize dates

+ Add dates

Sort: Earliest start date

1 date

4 Nights

Tue. **May 9** > Sat. **May 13**  
2023 2023

See 0 comments

**Next step: open voting**

Start collecting votes from travelers you've invited to the trip.

**Open voting**

**Skip voting**

## 8) defining destination

## Add a destination

X

Suggest a destination for your upcoming trip.

Where to?\*

Tuscany, Italy

X

Add a comment

Comment



Click the camera icon to add a custom photo

Save

1) create new plan

The screenshot shows a web browser window for the Planify.io manager at the URL [planify.io/manager/](https://planify.io/manager/). The page has a dark header bar with various navigation links like Tools, Training, SD, Tableau, ATS, DBs, Dev Dsk, uni, BI, career, Projects, Interviews, and mgmt material. Below the header, there's a message about a free trial version with 13 days left, and a red 'Upgrade' button. The main content area is a modal titled 'Create your plan / travel'. It contains fields for 'Name your plan / travel itinerary' (with placeholder 'Edit the name of your travel (ideally, less than 32 characters)'), 'Start date' (with placeholder 'Start time:'), a 'Description' text area (placeholder 'You can add a description of the travel here'), and a 'Tags' input field. At the bottom, there are sections for 'Add a cover picture' (with a dashed box and 'Upload' button), 'Add your logo' (with 'Premium only' text and a red 'UPGRADE' button), and a red 'Next' button.

2) plan details

Free trial version (day(s) left: 13) [Upgrade](#)

[Installation Stats](#)

EN Support Susanna V

< Dashboard / summer holidays Your data are automatically saved!

**PREVIEW**

**Agenda**

May 22, 2023 22/05/2023

walking tour hotel

**Activity name:** walking tour

**Type of activity:** Free Time

**Short description:** Add a short description

**Add a picture:** + Upload

**Long description:** You can add a long description of the activity here

**Start time:** 22/05/20 16:17 **End time:** 22/05/20 16:17

Use 24 hours format Use 24 hours format

**Address:** Type the name ... **Additional address:** Type the name ...

Add Website URL [Add Website...](#)

**CONTRIBUTORS / GUIDES / SPEAKERS**

[Create and link an activity contributor](#)

**ADD A DOCUMENT**

Click or drag file to this area to upload  
Support for a single or bulk upload. Avoid uploading company data or other confidential files

**PDF PREVIEW**

May 22, 2023

**summer holidays**

**P Program:**

Monday 22nd May 2023

16:17 - walking tour  
16:17 - Free Time

16:17 - hotel

Pilotplans.com

1) home – plans list

## Upcoming Trips

[Create a trip](#)

Create new trip

## Past Trips

Oh no! You have no past trips 😞

2) plan details

app.pilotplans.com/trips/-NVyZ0JFVwmd5elZPlyq/planner

Tools Training SD Tableau ATS DBs Dev Dsk uni BI career Projects Interviews mgmt material Other bookmark

**summer holidays** + Add Dates

Trip Plan Notes Files Saved

+ Add a trip photo

Visas & Requirements Export Your Trip Imported Files

**Lisbon** May 21st - 26th

Hôtel de Ville

+ Accommodation

New Heading May 22nd

bunjee

+ Activity + Heading

Departure Arrival

**Porto**

+ Add Dates

Add accommodation

New Heading + Add Dates

new activity

How disappointed would you be if you could no longer use Pilot?

- Very Disappointed
- Somewhat Disappointed
- Not Disappointed

*TripIt*

1) new plan

## Edit Trip Info

Add a trip manually below or forward your confirmation emails to [plans@tripit.com](mailto:plans@tripit.com), and we'll create the trip for you.

Trip Name

Destination City \*

×

Start Date \*

End Date \*



[Change Photo](#)

[Crop Photo](#)

I am a traveler on this trip

[Cancel](#)

[Save](#)

The screenshot shows the Plan Harmony mobile application interface. At the top left is the app logo 'Plan Harmony'. On the top right are notification and user profile icons. The main header is 'Upcoming Trips'. Below the header is a teal-colored button labeled 'Start Planning a New Trip'. A small globe icon is positioned above the button. A quote below the button reads: 'It doesn't matter where you're going, it's who you have beside you'. At the bottom of the screen is a teal-colored 'Get Planning →' button.

The screenshot shows the 'Start Planning' screen of the Plan Harmony mobile application. It features a large teal-colored input field with the placeholder 'lisbon'. Below this are two date inputs: '05/24/2023' and '05/31/2023'. Underneath the dates is a dropdown menu set to 'Europe/Berlin'. A note below the dropdown says: 'Timezones are hard. Adding the right timezones will make viewing your trip much easier'. At the bottom of the screen is a teal-colored 'Get Planning →' button.

2) new event in plan

The screenshot shows the Plan Harmony mobile application interface. On the left, a sidebar menu includes options like 'New Trip', 'All Trips', 'Calendar' (selected), 'Experiences', 'Budget', 'Checklists', 'Map', 'Settings', and 'Send Feedback!'. Below the sidebar are social sharing icons for Instagram, Facebook, Twitter, LinkedIn, and Email.

The main area displays a trip summary for 'lisbon' (Plan H) on 'May 24'. A 'Create Event' modal is open, titled 'Create Event' with a search bar containing 'hotel checkin'. Below the search bar is a row of icons for different event types: Home, Work, Airplane, Car, Train, Bus, Bike, Cabin, and a trash bin.

The event creation form includes the following fields:

- Guests attending this event:** Shows 'SV' (Selected Guests).
- Event Dates and Times:** Set from '05/24/2023 12:00 AM' to '05/24/2023 1:00 AM' in Europe/Berlin. There is a toggle for 'All Day Event?'.
- Event Type:** A toggle for 'Event optional or pending decision' with the note 'A great way to make group decisions!'.
- Description & Notes:** A text input field for adding a description and notes.
- Attachments:** A note 'Please save this event before uploading attachments' with a link to 'Upgrade Now! →'.
- Address:** An input field for entering an address.
- Confirmation Code:** An input field for entering a confirmation code.
- URL:** An input field for entering a URL.
- Budget:** A section showing '\$ 100.00' and a 'Split Between' dropdown menu. It also includes a '+ Track Additional Cost' button and a note to 'Upgrade this trip to Plan Harmony Pro for additional budget tracking'.

At the bottom of the modal are two buttons: 'Save Event' and 'Delete this Event'. To the right of the modal is a blue circular icon with a white speech bubble and a person icon.

3) premium plan

## Plan Harmony Pro

Upgrade this trip to Plan Harmony Pro for advanced features

- ✓ Sync your trip to your favorite calendar
- ✓ Robust budget tools (multiple costs, export and more!)
- ✓ Unlimited checklists
- ✓ Unlimited file attachments

\$3.99

Upgrade Now →

New pro features coming all the time!

### Invite Guests

- Planning a trip with someone is way more fun!  
Inviting guests to collaborate, vote, and create events is what makes this a way better experience! Any guest added to the trip has the ability to invite someone.



Susanna V  
susy.venta.93@hotmail.it



Susanna Ve  
susanna.ventafredda@gmail.com

Email address

Comma separated list of email addresses

Invite

### Plan Harmony Pro

#### Calendar Sync

Send this trip to your favorite Calendar. It's a great way to view your trip if internet connectivity is a little spotty.

Upgrade this trip to Plan Harmony Pro to unlock this feature

**Note:** Anyone you share this link with will be able to have read-only access to your trip's information and can take up to 24 hours to refresh. Choose your application from the list below for instructions on how to subscribe to the iCal feed



### Plan Harmony Pro

#### Export Events

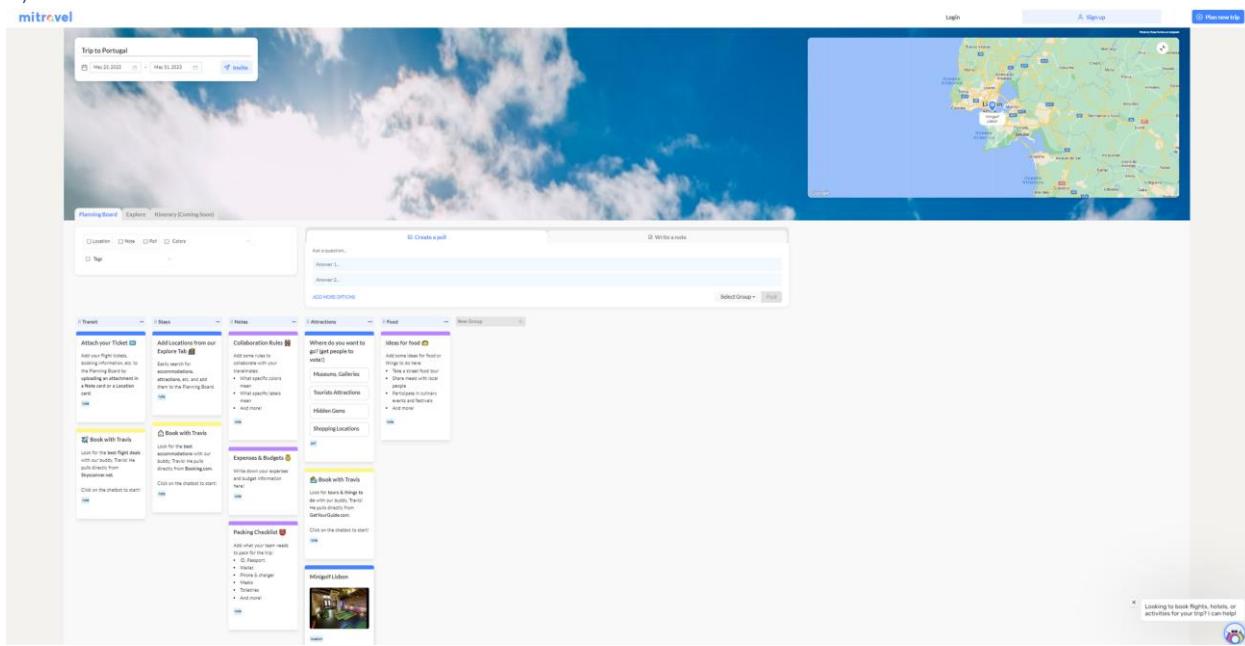
Export all of your events to a csv file and open it in excel or google sheets!

Upgrade this trip to Plan Harmony Pro to unlock this feature



Mitravelapp

## 1) home view



## Appendix B – Functional requirements and acceptance criteria

n.	Feature	Goal	User Story	Requirement	Priority
1	Experience main functionality without logging in	Allowing customers to explore website functionality and gain customer base	As a potentially interested new user, I want to be able to understand what the website allows to do without overcommitting from the start: without registering an account and without committing to pay anything.	<p>The system must allow unauthenticated users to visit most pages of the website and interact with main features:</p> <ul style="list-style-type: none"> <li>• seeing existing mock-up plan</li> <li>• creating a plan (but not saving it)</li> <li>• adding event to plan (but not saving it)</li> </ul> <p>The system must clearly display a message that they need to log in to create personal</p>	High

				plan and that they are only viewing in demo mode – warning div below navbar.	
2	User registration	Allowing new users to sign up to our website, in order to start creating and saving plans, and be able to invite friends.	As a new user that likes the tool's functionality, I want to be able to sign up by providing the minimum personal information possible and in the fastest way possible, in order to start using the tool	The system must allow new users to create a new account by providing email, name and surname.	High
3	User login	Allowing existing users to access their profile and saved trip plans by providing email and password	As an existing user that has already created an account, I want to be able to log in to my existing account to access all information saved so far	The system must allow existing users to access their account by providing email and password	High
4	Password reset	Allowing users to reset their password if they forgot it	As a user who created an account in the past, I want to be able to access my account even if I forgot my password, by requesting a change of password.	The system must allow users who forgot their password to enter their email address and request that a link to change their password is sent to their inbox.	High
5	Display clear user agreement and data privacy documentation	Reassuring customers on data protection policies adopted by the website and how their data is used and stored	As a new user, I want to know how my data will be used by the website before creating a new profile	The system must link documentation on user agreement and data privacy that the new user must accept	High

				before registering a new account	
6	Logout	Allowing users to log out of their account and access as a different account	As an authenticated user, I want to be able to log out of my account	The system must allow logged in users to log out of their account	High
7	Delete account	Allowing users to delete their account once they no longer need to use our tool	As an existing user that is no longer interested in using the tool, I want to be able to delete my account	The system must allow authenticated users to delete their existing account	High
8	Communicating to developers / product owners	Allowing authenticated and unauthenticated users to report bugs, submit feature requests, and communicate any other issue/opportunity to app developer. Helps deescalate problems if they arise, by being aware of them and tackling them as soon as possible	As a user experiencing an issue / having an idea to improve the app, I want to be able to communicate my issue / idea to the app owners	The system must allow authenticated and unauthenticated users to fill in a form and submit it to communicate with app owners	High
9	Create a new plan	Allowing existing users to start planning a trip	As an authenticated user, I want to be able to create a trip plan	<ul style="list-style-type: none"> <li>The system must allow authenticated users to click on “New trip” from their home page.</li> <li>The system must allow user to fill in a form containing trip title, date preferences, number of days available to travel</li> </ul>	High

				<ul style="list-style-type: none"> <li>• The system must allow users to submit the form and save data into the database</li> <li>• The system must confirm to the user that their information is saved and available from the home page. Allow users to acknowledge with ok.</li> <li>• The system redirects to home page once users click ok</li> </ul>	
10	Abandon creation of new plan	Allowing to abort generation of new plan	As an authenticated user who is in the process of filling in the form to create a new plan, I want to make sure that I can interrupt and exit this process at any time	<ul style="list-style-type: none"> <li>• The system must display a cancel button at the bottom of the form to allow users to quit form filling process.</li> </ul>	High
11	Display existing plans on the home page	Allowing authenticated users to view all the plans they have created or they are collaborating on	As an authenticated user, I want to be able to have all the plans I created available in one place that is easy to find and be able to access such plans. I also want to be able to filter existing	<ul style="list-style-type: none"> <li>• The system must allow authenticated users to view all their plans on the home page</li> <li>• The system must allow user to filter trips through a dropdown: upcoming trips, past</li> </ul>	High

			plans I created to easily find them	trips. By default, the system should display only upcoming trips, order by last modification date	
12	Delete plans from the home page	Allowing authenticated users to remove unwanted plans quickly and easily	As an authenticated user, I want to be able to delete any of the plans I created	<ul style="list-style-type: none"> <li>The system must display a delete icon on top of each plan card.</li> <li>The system must request user confirmation with a popup modal whenever a user clicks on such delete icon.</li> <li>The system must delete from the database related entries for the deleted plan</li> </ul>	High
13	View details of existing plan	Allowing users to see details of existing plan	As an authenticated user, I want to be able to open any of the plans linked on the home page and view their details	The system must allow users to click on any plan displayed on the home page and open a new page containing the details of the selected plan	High
14	View/edit participants of plan	Adding friends to a trip plan	As an authenticated user, I want to be able to add one or more friends to a given trip plan	<p>The system must allow authenticated users to:</p> <ul style="list-style-type: none"> <li>Click on the 'add friends' button in the participants</li> </ul>	High

				<ul style="list-style-type: none"> <li>tab of the trip details</li> <li>• Choose between sharing a link with friends, entering friend's email address, choose an existing friend</li> <li>• Allow users that receive the sign up link to register a new account and confirm they want to join the trip and become friends with the user</li> <li>• If user already has an account, just join the trip and add to friends</li> </ul>	
15	View / edit itinerary of plan	Viewing the content of a plan or modifying it	As an authenticated user, I want to be able to open any of the plans and view the details of their itinerary. I also want to add items to the itinerary	<p>The system must allow authenticated users to:</p> <ul style="list-style-type: none"> <li>• Access the itinerary tab of the trip view page</li> <li>• Open existing events to see their content</li> <li>• Modify the content of existing events they created</li> <li>• Comment on existing events</li> <li>• Vote on existing events that require user vote</li> </ul>	High

				<ul style="list-style-type: none"> <li>• Filter events by category</li> <li>• Add new event and save it to database.</li> <li>Refresh itinerary page with new event</li> </ul>	
16	View / edit polls for the plan	Allowing users to create custom polls to facilitate group decisions.	As an authenticated user, I want to view polls I created or that require my attention and also create new polls	<p>The system must allow authenticated users to:</p> <ul style="list-style-type: none"> <li>• Access the polls tab from the trip view page</li> <li>• See existing polls, ordered by action required and creation date</li> <li>• Open details of existing polls</li> <li>• Create new polls</li> </ul>	High
17	delete plan from plan details	Allowing users to delete a plan if no longer needed	As an authenticated user, I want to be able to delete a plan from the plan details view	The system must allow authenticated users to delete a plan form the plan view page	High
18	Remove myself from plan	Allowing to remove a user from an existing plan they have not created and they've been added to	As an authenticated user, I want to be able to remove myself from a plan if for any reason I no longer want to participate	The system must allow authenticated users to remove themselves from a plan by clicking on the bin icon within the participants tab of the plan view page	High
19	Edit trip name	Allow users to change their mind on a plan name and correct mistakes	As an authenticated user, I want to be able to modify a	The system must allow authenticated users to rename a	High

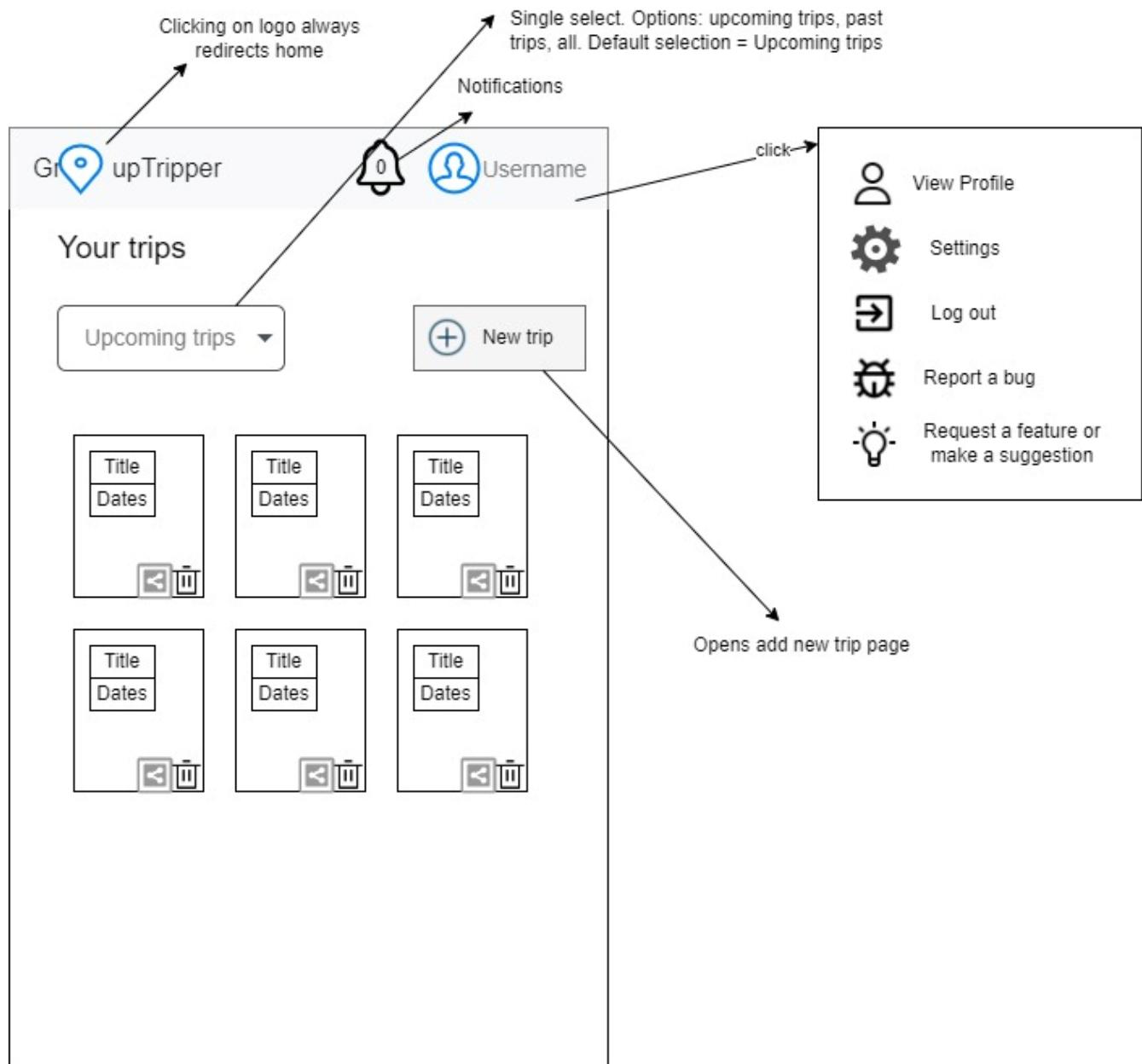
			plan name after I created a plan	plan from the plan view page	
20	View / edit personal info	Allow users to edit their name and surname after they have created their profile	As an authenticated user, I want to be able to go to my profile page and edit my name, surname and profile picture	The system must allow authenticated users to: <ul style="list-style-type: none"> <li>• edit their profile picture</li> <li>• edit their name</li> <li>• edit their surname</li> </ul>	High
21	View / edit friends	Allow users to see list of friends associated to their profile	As an authenticated user, I want to be able to go to my profile's 'friends' page and see all friends linked to my profile, remove or add friends	The system must allow authenticated users to: <ul style="list-style-type: none"> <li>• view existing friends</li> <li>• remove existing friends</li> <li>• view pending friends</li> <li>• add friends by sharing a link or entering their email address</li> </ul>	High
22	Change language	Allow users from different nationalities to use the site. Increase accessibility.	As a non-English-speaking user, I want to be able to still use the website and understand its content	The system must allow users to specify the language they prefer in a visible position on the navbar. The system must translate all content upon selection of a language	Medium
23	Map view	Allow users to see their itinerary on a map	As an authenticated user, I want to be able to view all items of my itinerary that have an address on a map	The system must display a map with pins for each location listed as a desired itinerary	Medium

	Allow to make plans public	Allow any user to browse already made plans for their destinations and copy existing templates / vote on them	As a user, I want to be able to see if there are existing plans for the trip I am about to plan and use it as a template	<ul style="list-style-type: none"> <li>• The system must allow users to make their plans public</li> <li>• The system must allow users to browse existing public plans filtering by location and days</li> <li>• The system must allow to copy an existing template to new and edit it</li> </ul>	Low
--	----------------------------	---	--	---	-----

## Appendix C – Medium-fidelity prototypes

### C.1 - First iteration

#### Home page



## Add new trip

upTripper

**Title**

This can be a generic title to identify your trip. For example: "Summer holidays at the seaside with office folks"

**Dates**

Have all participants already agreed on the dates? Yes

Select start date: No date selected

Select end date: No date selected

Add more dates when you're available

How many working days are you willing to travel? 1

Cancel

If you select 'YES', participants will only be asked if they're ok with selected dates. If 'NO' is selected, all participants will be asked their dates availabilities in order to find common availability

opens modal

open modal

opens modal with calendar

only visible if user selects 'No' in the drop-down menu 'Do you already know the dates of your trip?'. Adds a div with additional dates. and possibility to add more dates. At least one pair of start and end dates must be specified.

OK

closes the modal

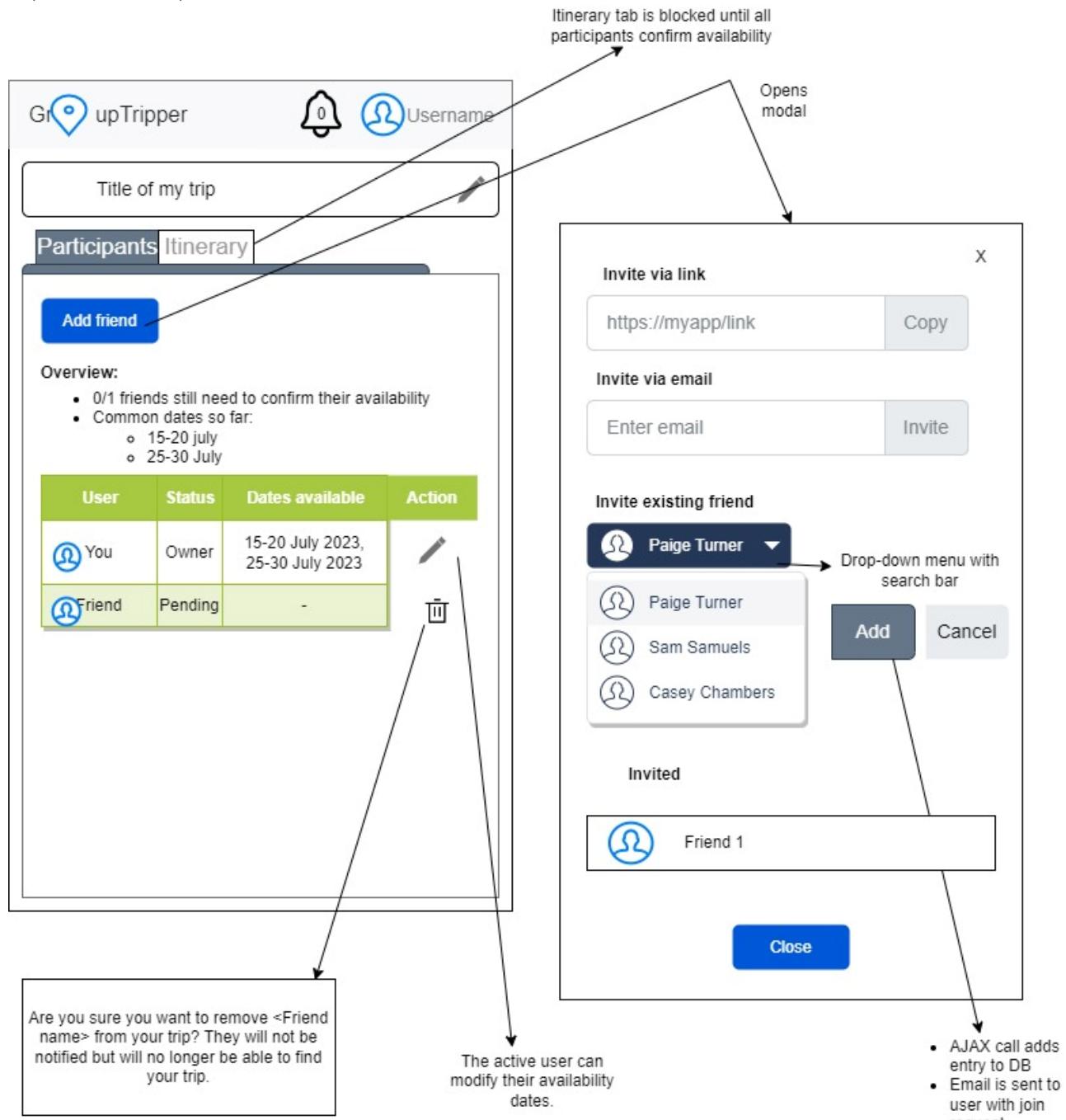
<	October 2014	>				
Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

Select start date: No date

Select end date: No date

Add more dates when you're available

## Trip view – Participants



## Trip view - Itinerary

The screenshot shows the UpTripper app interface. At the top, there's a header with the logo and the text "upTripper". To the right are icons for notifications (0) and "Username". Below the header, there's a search bar with the placeholder "Title of my trip" and a pencil icon. Underneath the search bar, there are two tabs: "Participants" and "Itinerary" (which is selected). A date range "Start date: 15 July 2023 - End date: 30 July 2023" is displayed. Below the date range is a button labeled "+ Add event". A box labeled "Open polls" is shown. A list of items follows:

- Event name - start date end date
- Status: awaiting votes / to book / booked
- Possible actions: edit, delete, comment, view votes

**New Event**

Event type

- Poll
- City
- Accommodation
- Transportation
- Attraction / Event
- Food and drinks
- Task / Todo reminder

Example fields:

- Name / Description
- Start and end date
- Timezone
- Address
- URL
- Cost and how the cost is split - only for pro membership
- Description and notes
- Attach files - only for pro membership
- participants: allows to unselect some participants
- Ask participants: Will you join / participate / stay here? Yes / no
- Require answer from all participants? Yes / no
- Automatic status field

Dynamic form: fields change depending on above selection

**Poll**

Question

Answer

Option 1

Option 2

Option 3 Delete

+

How many options can participants choose? 1 ▼

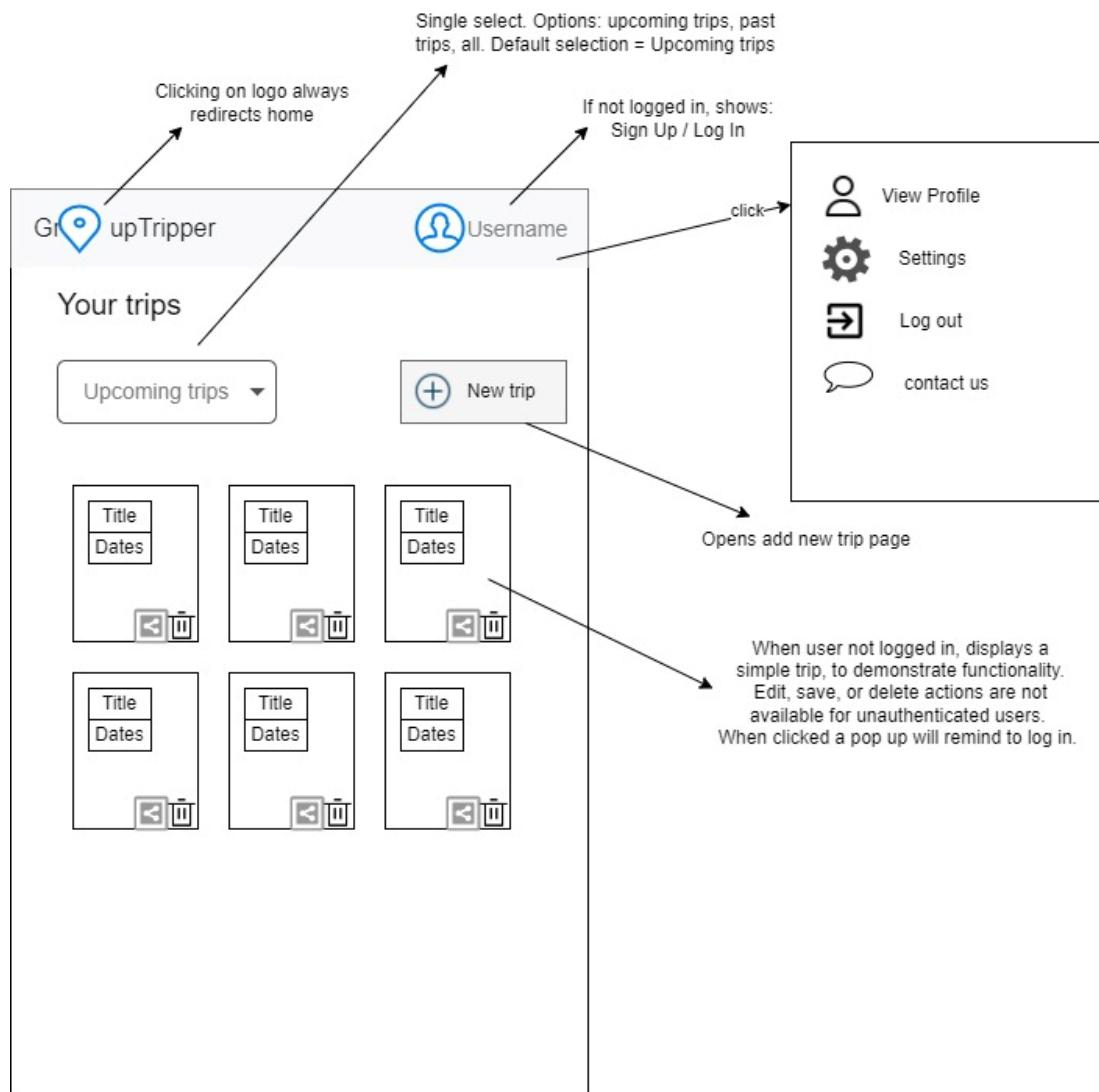
Participants

By default, lists all trip participants. Option to remove people with switch button Off

Cancel Send

## C.2 - Second iteration

### Home page



## Add new trip

upTripper

Username

Title

This can be a generic title to identify your trip. For example: "Summer holidays at the seaside with office folks"

OK

closes the modal

Dates

Have all participants already agreed on the dates?

Yes

If you select 'YES', participants will only be asked if they're ok with selected dates. If 'NO' is selected, all participants will be asked their dates availabilities in order to find common availability

Select start date

No date selected

Select end date

No date selected

Add more dates when you're available

+

Cancel

Save

only visible if user selects 'No' in the drop-down menu 'Do you already know the dates of your trip?'. Adds a div with additional dates. and possibility to add more dates. At least one pair of start and end dates must be specified.

<	October 2014	>				
Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

Select start date

No date

Select end date

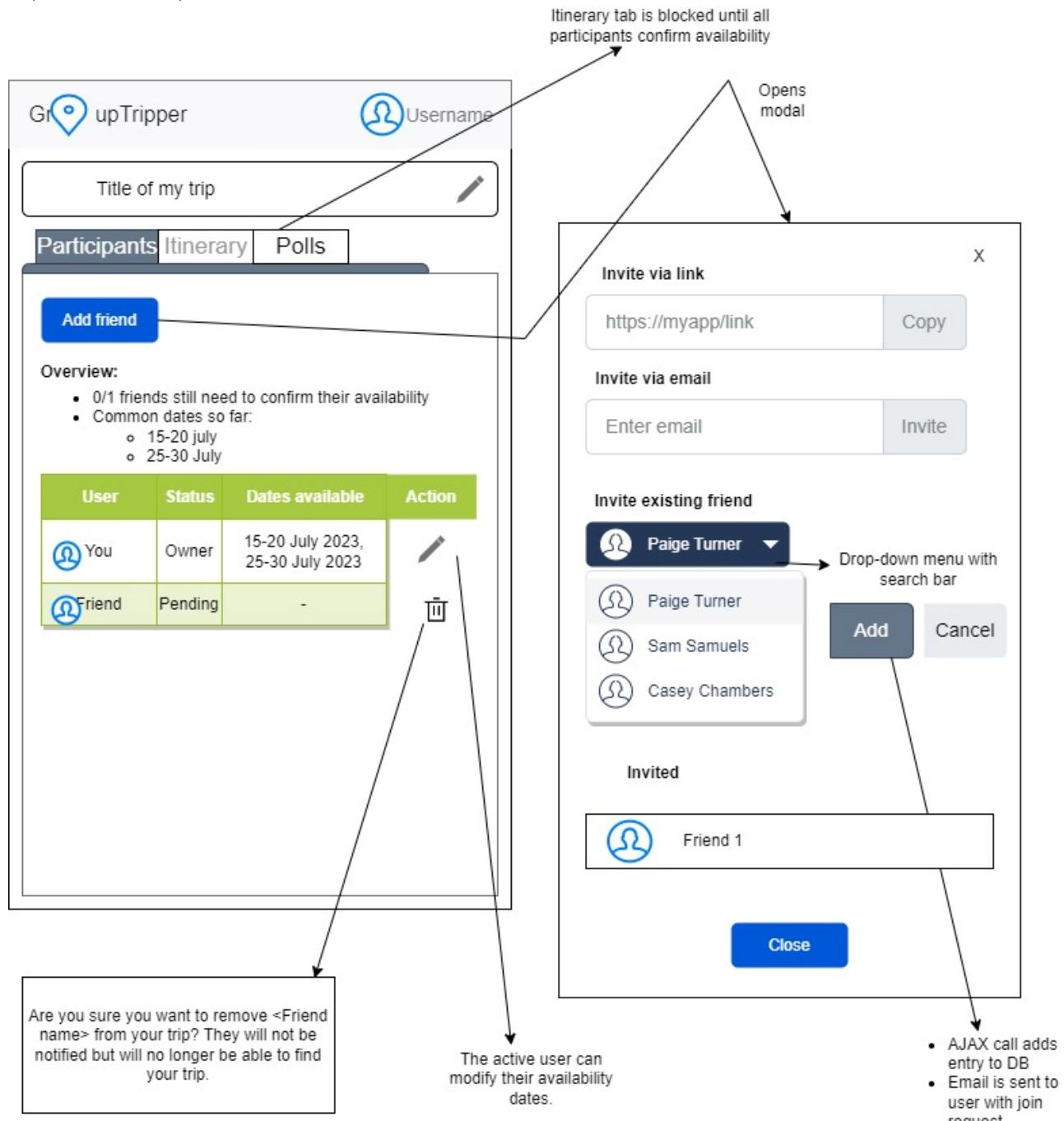
No date

+

Add more dates when you're available

+

## Trip view – Participants



Are you sure you want to remove <Friend name> from your trip? They will not be notified but will no longer be able to find your trip.

The active user can modify their availability dates.

Itinerary tab is blocked until all participants confirm availability

Opens modal

**Invite via link**

<https://myapp/link>

**Invite via email**

Enter email

**Invite existing friend**

Paige Turner Drop-down menu with search bar

Paige Turner  
Sam Samuels  
Casey Chambers

Add Cancel

**Invited**

Friend 1

Close

- AJAX call adds entry to DB
- Email is sent to user with join request

## Trip view - Itinerary

Same as Edit view

The screenshot shows the 'Itinerary' tab selected in the navigation bar. A search bar at the top contains the placeholder 'Title of my trip'. Below it, there are tabs for 'Participants', 'Itinerary' (which is active), and 'Polls'. A date range is displayed: 'Start date: 15 July 2023 - End date: 30 July 2023'. A button labeled '+ Add event' is present. The main area lists an event with the title '<Event name>', status 'awaiting votes, to book / booked', and a note 'start date end date'. It also shows '2 comments'. A dropdown menu 'All categories' is visible.

The screenshot shows the 'New Event' dialog. It includes fields for 'Event type' (dropdown), 'Name / Description' (text input), 'Start date' (calendar), 'End date' (calendar), 'Timezone' (dropdown), 'Address' (text input), 'Description and notes' (text input), 'Upload attachments' (file input), 'Cost' (input), 'Currency' (dropdown), 'Participants' (list box with 'Friend 3'), and 'Ask participants if they will join' (checkbox). Buttons for 'Cancel' and 'Save' are at the bottom. A sidebar on the right lists event types: City, Accommodation, Transportation, Attraction / Event, Food and drinks, Task / Todo reminder. A calendar interface for selecting dates is shown, with the 24th highlighted in red. A note indicates 'Only for pro membership (not in MVP)'.

Events are ordered by date, displaying cities first

On click, opens dedicated page (same as edit view, but with no editable fields. Displays buttons to Close, Edit (only for task owner), Delete (only for task owner)). Displays section with comments at the end, which also allows to add comments.

## Trip view - Polls

The diagram illustrates the user flow from the 'Trip view - Polls' screen to the 'Poll' creation screen.

**Trip view - Polls Screen:**

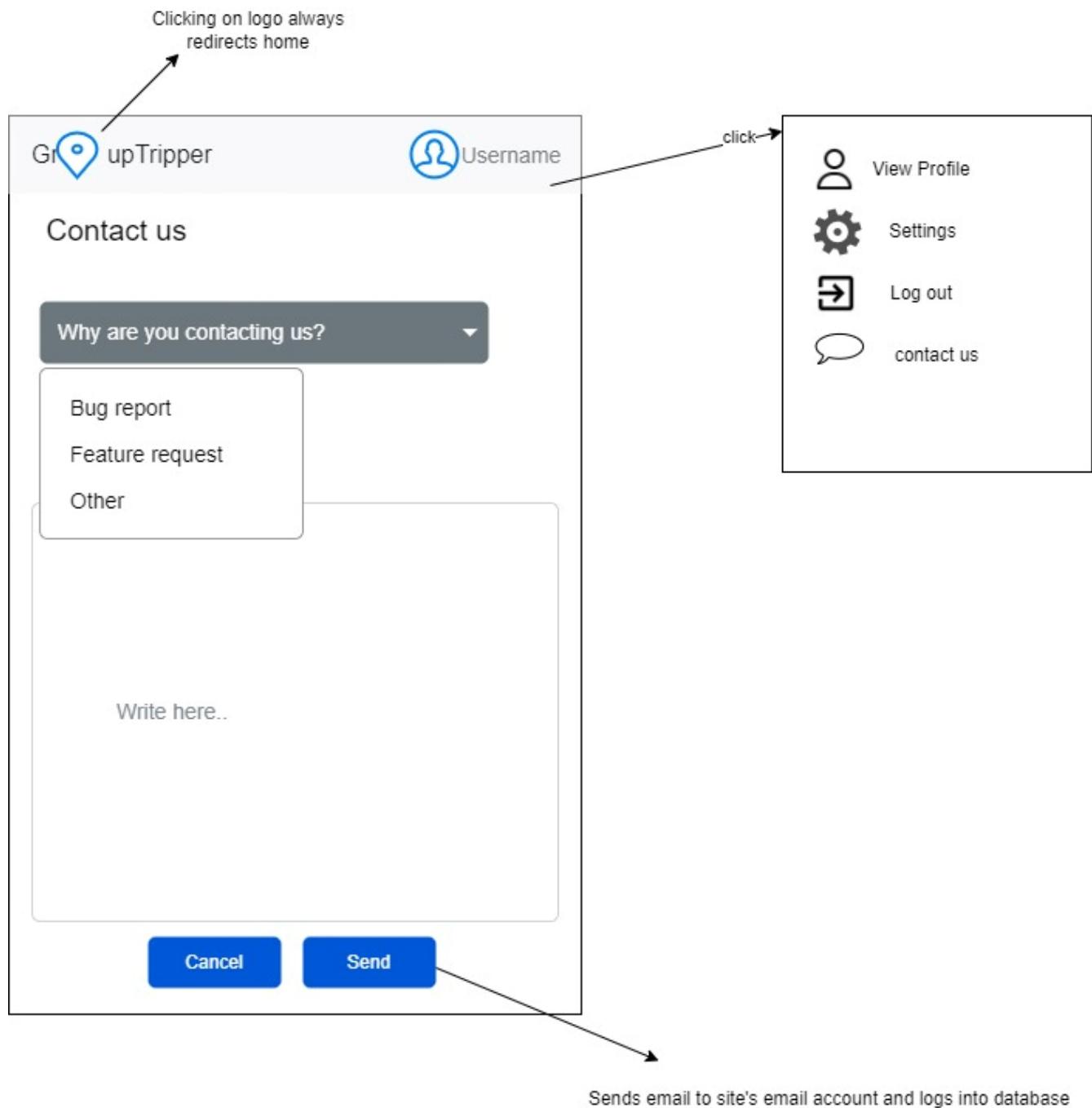
- Top navigation: GroupUpTripper logo and Username.
- Title bar: Title of my trip with a pencil icon.
- Tab bar: Participants, Itinerary, and Polls (selected).
- Buttons: New poll (with a plus sign) and Your polls.
- Filter: Polls currently open ▾ (dropdown menu showing Options: open, closed, all).
- Poll card: Poll question "You still need to vote!" (with an exclamation mark icon), Votes: 1/3, and a View button.

**Poll Creation Screen:**

- Section title: Poll.
- Form fields:
  - Question: Input field.
  - Answer: Input field containing Option 1.
  - Answer: Input field containing Option 2.
  - Answer: Input field containing Option 3.
  - Action: Delete icon (trash can).
- Add button: A blue button with a plus sign (+).
- Setting: How many options can participants choose? (set to 1).
- Participants section: By default, lists all trip participants. Option to remove people with switch button.
- Action buttons: Cancel and Send.

Only shows polls that you created or that require your vote.  
display first polls with open action item

## Contact us



## Settings

Clicking on logo always  
redirects home

upTripper

Username

## Settings

Email notifications

Users accept your friendship request

You receive new friendship requests

Poll requires your input

Your friends request your input

New comments are added to your plans

New events are added to your plans

View Profile

Settings

Log out

contact us

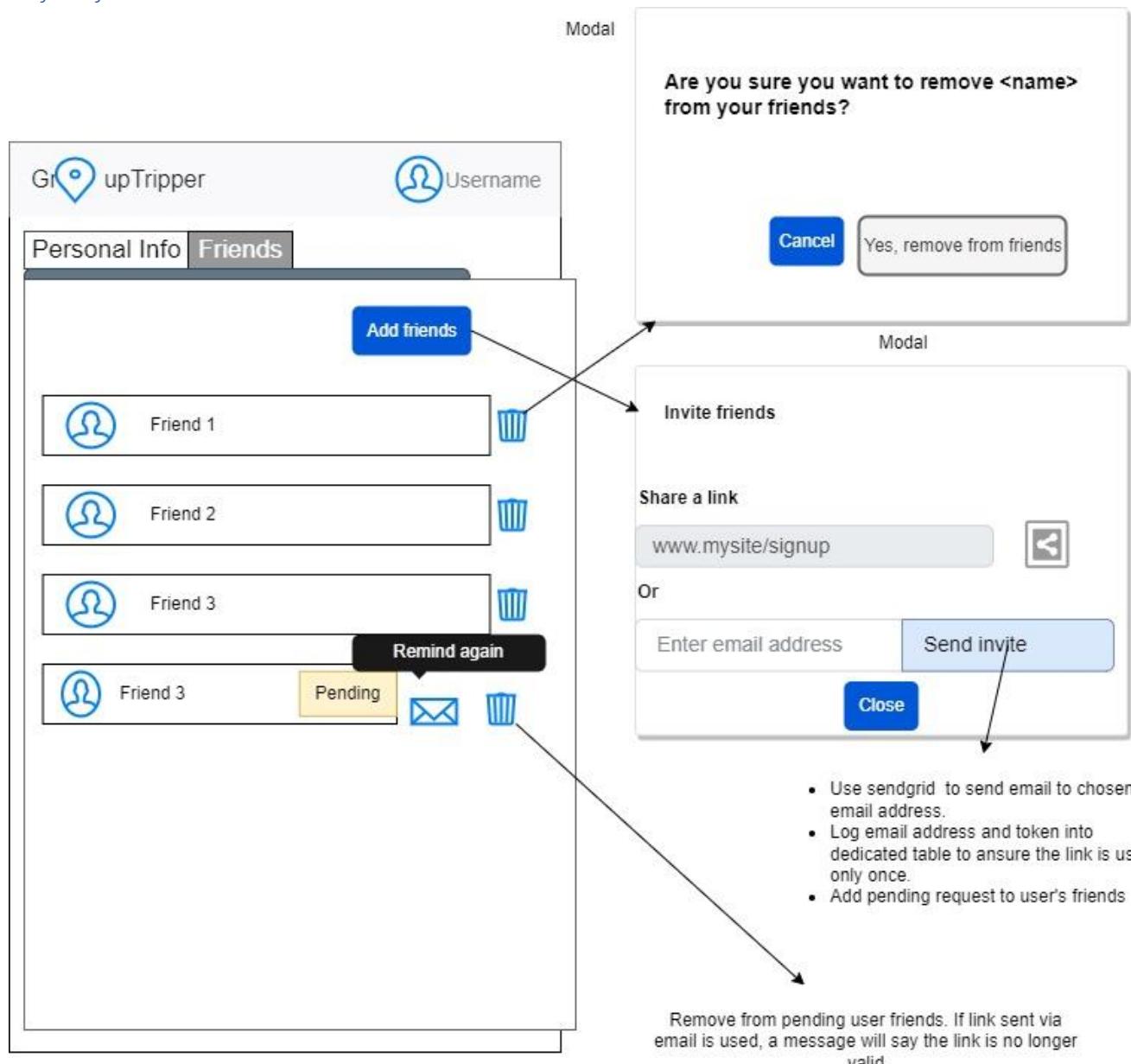
Profile – personal information

The screenshot shows a user profile editing interface. At the top left is the 'upTripper' logo. To its right is a placeholder for a profile picture with the text 'Username'. Below the logo is a navigation bar with 'Personal Info' (highlighted in blue) and 'Friends' tabs. A large circular placeholder for a profile picture is centered below the tabs. To the right of the placeholder is a small blue pencil icon. Below the placeholder are two input fields: 'First name' containing 'Somename' and 'Last name' containing 'Somesurname'. A blue 'Save' button is located at the bottom of the form area.

Initially these are read only fields. When user clicks on edit icon, page reloads and allows modifications

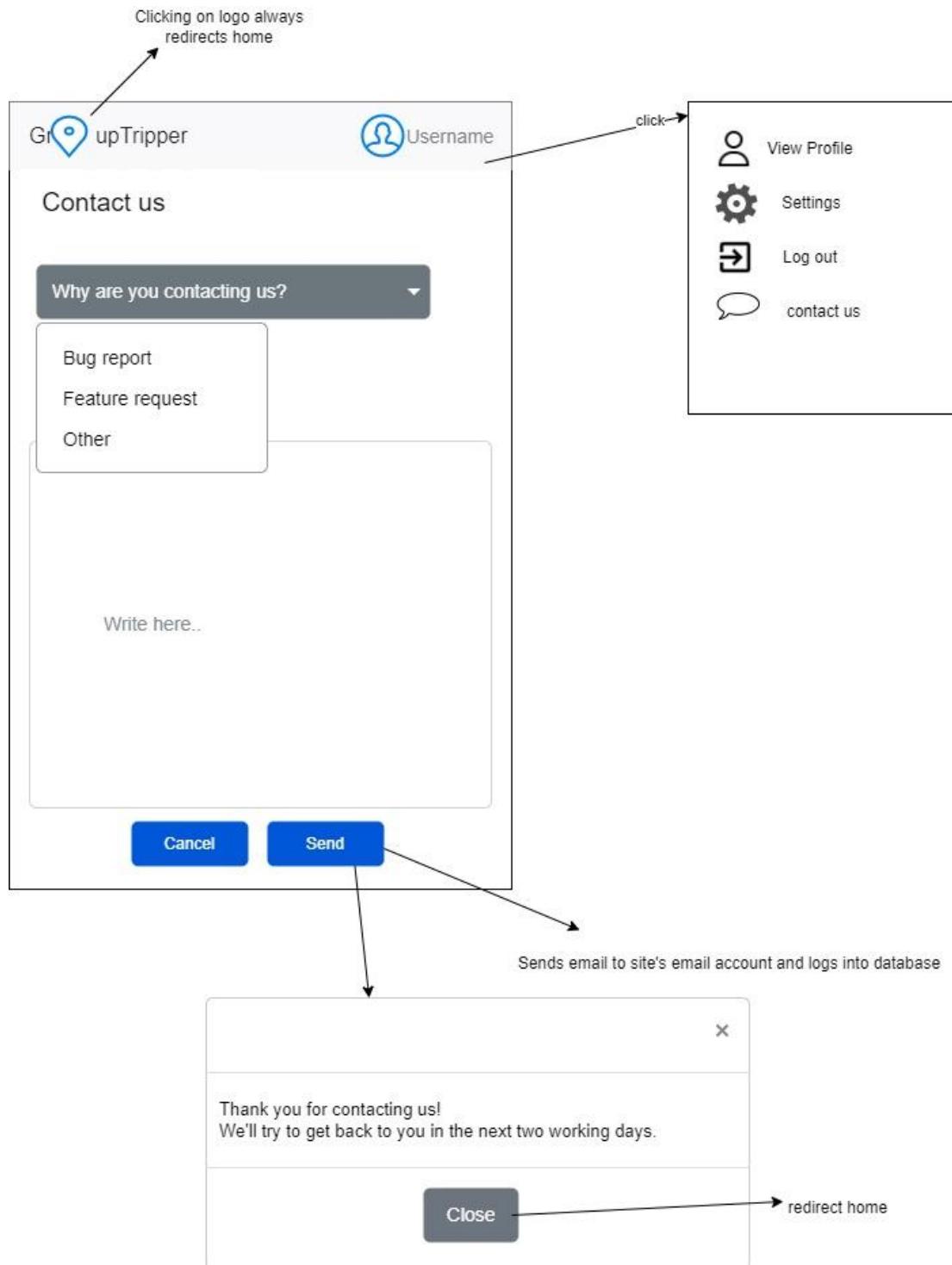
Only visible on edit mode

## Profile – friends



### C.3 - Third iteration

#### Contact us



## Settings

Clicking on logo always  
redirects home

upTripper

Username

Settings

Email notifications

Users accept your friendship request

You receive new friendship requests

Poll requires your input

Your friends request your input

New comments are added to your plans

New events are added to your plans

All participants finish voting on an item / poll

View Profile

Settings

Log out

contact us

## Add new trip

This can be a generic title to identify your trip. For example: "Summer holidays at the seaside with office folks"

OK

closes the modal

opens modal

Dates

Ask all participants when they're available before finalizing dates

Select start date No date selected

Select end date No date selected

Add more dates when you're available +

How many working days are you willing to travel? 1

How many days are you willing to travel in total? 1

Cancel Save

other option

Propose dates and ask participants to accept or decline

opens modal with calendar

	<	October 2014	>			
Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

only visible if user selects 'No' in the drop-down menu 'Do you already know the dates of your trip?'. Adds a div with additional dates. and possibility to add more dates. At least one pair of start and end dates must be specified.

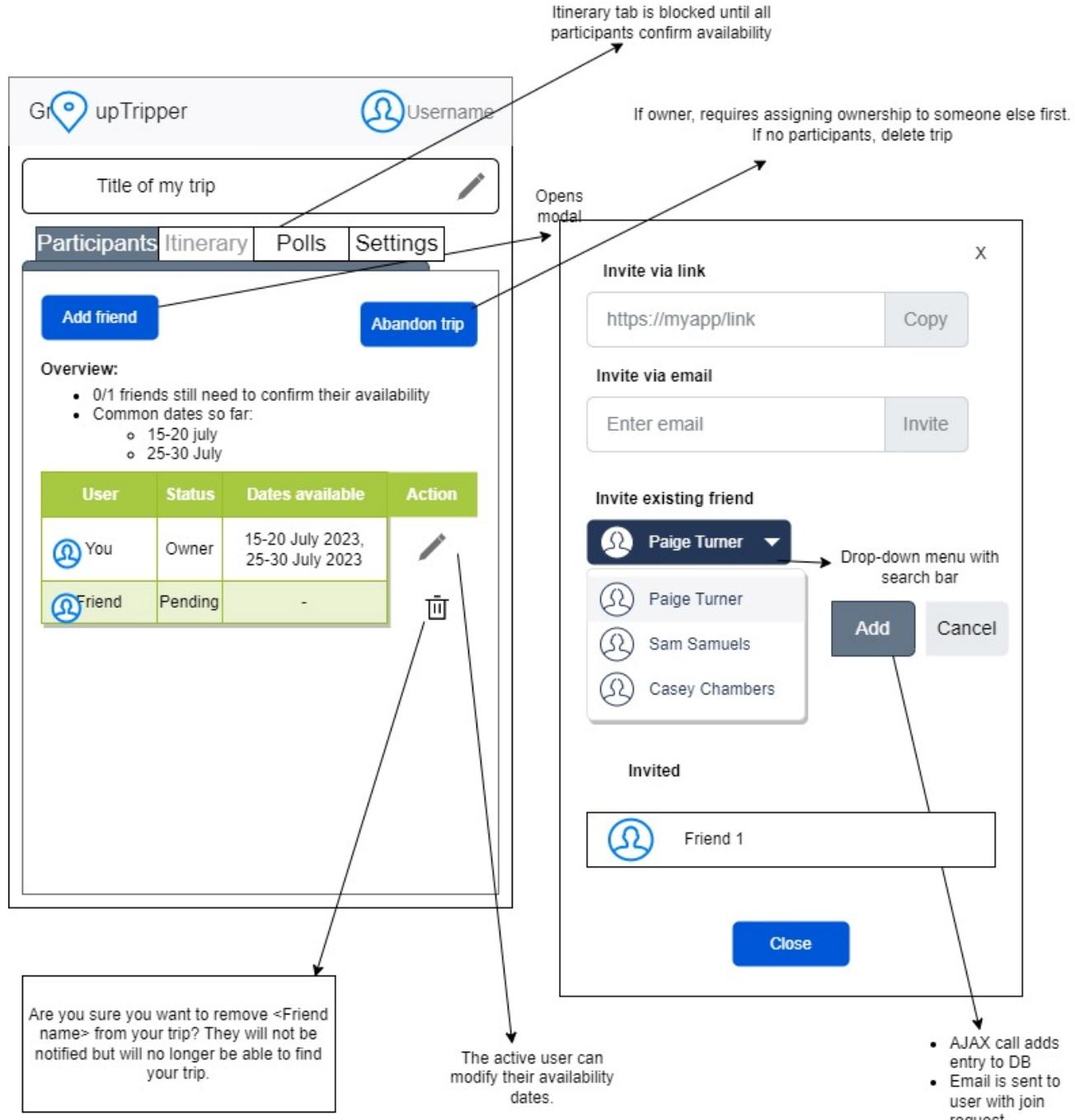
Select start date No date

Select end date No date

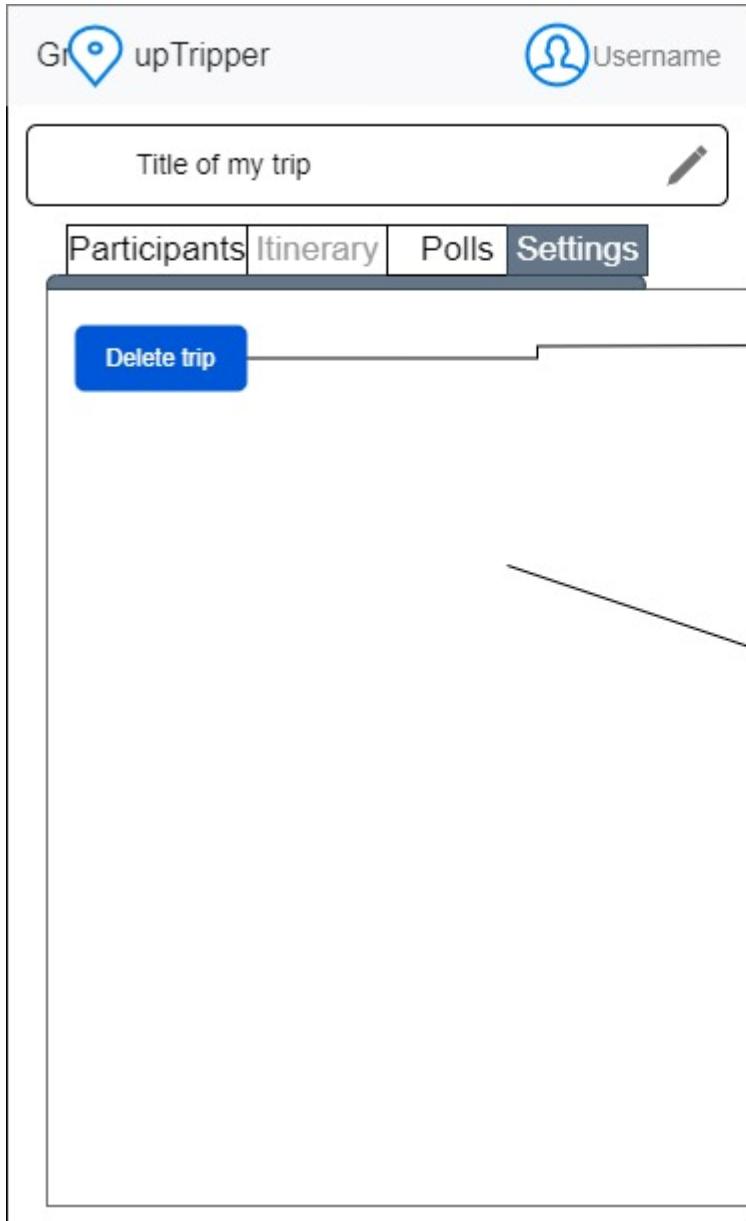
Add more dates when you're available +

87

## Trip view – Participants



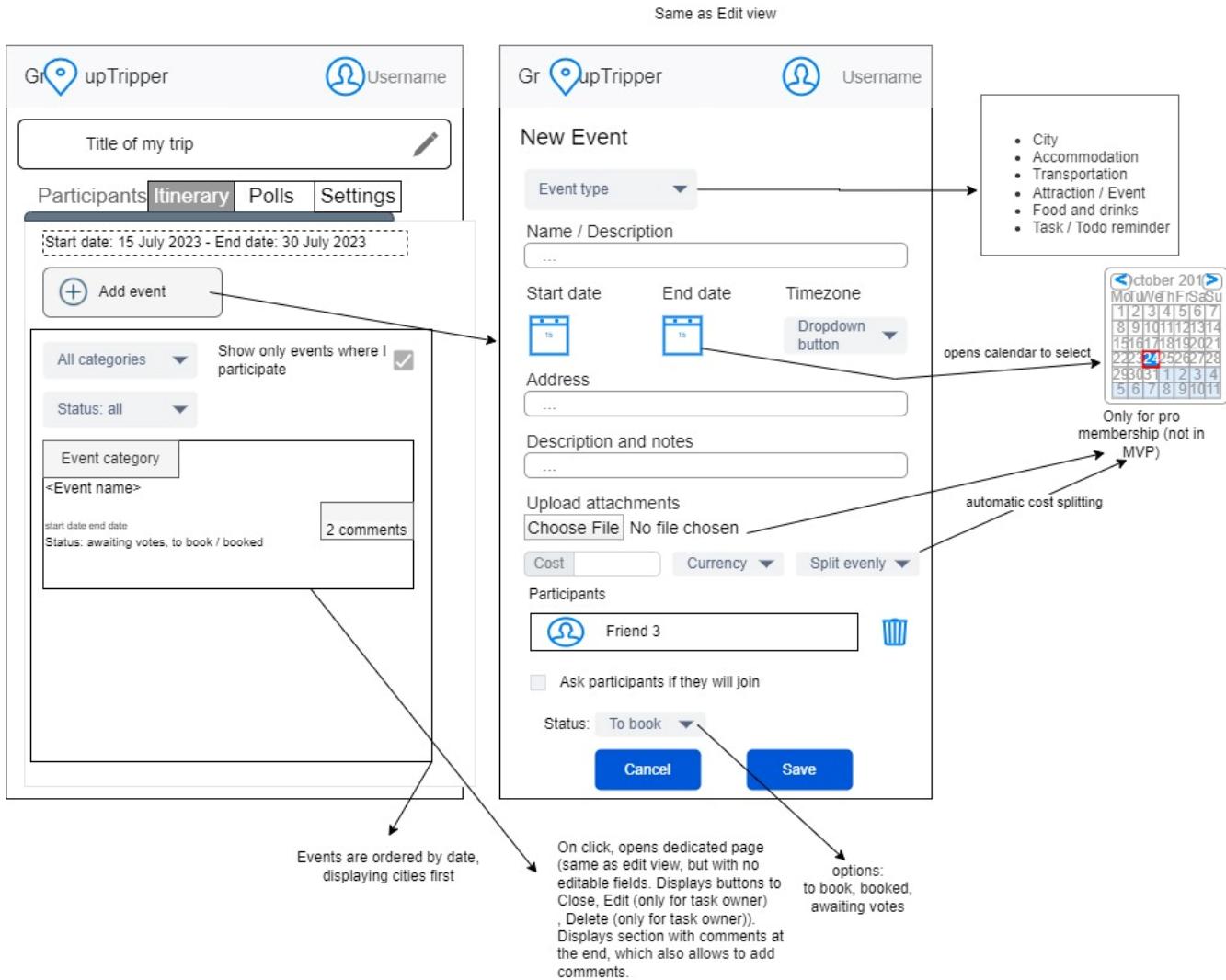
## Trip view – Settings



Triggers modal asking confirmation, then modal to confirm deletion, then redirects home

section will contain options on privacy, after MVP launch (to make plans public)

## Trip view – Itinerary



## Appendix D – Email sending costs

Email platform	Cost	Chosen
Google – SendGrid Email API [61]	<ul style="list-style-type: none"> <li>12,000 free / month</li> <li>50,000: 19.95 USD / month</li> <li>100,000: 34.95 USD / month</li> </ul>	Yes, because cheapest
Mailjet [62]	<ul style="list-style-type: none"> <li>6,000 free / month. 200/day max</li> <li>15,000: 15 USD / month</li> <li>Price for more than 15,000 only on demand</li> </ul>	
Mailgun [63]	<ul style="list-style-type: none"> <li>5,000 free / month. Trial only for 1 month.</li> <li>50,000: 35 USD / month</li> <li>100,000: 90 USD / month</li> </ul>	

## Appendix E – Addresses autocomplete costs

Service	Cost	Comments	Chosen
Geoapify [49] - <a href="#">Demo</a>	<ul style="list-style-type: none"> <li>• 3,000 requests / day for each type of request for free. Max 5 requests/second</li> <li>• 10,000 requests / day for each type of request for 49 €. Max 12 requests/second</li> <li>• 25,000 requests / day for each type of request for 89 €. Max 15 requests/second</li> <li>• 50,000 requests / day for each type of request for 149 €. Max 20 requests/second</li> </ul>	<ul style="list-style-type: none"> <li>• Clearly differentiates countries, cities, addresses in search results</li> <li>• Limit on requests per second</li> </ul>	<ul style="list-style-type: none"> <li>• No because request per second limitation</li> </ul>
Google Place Autocomplete API [50] - <a href="#">Demo</a>	<ul style="list-style-type: none"> <li>• 0.00283 USD per each request (0–100,000 requests / month) == 2.83 USD for 1000 requests</li> <li>• 100,001–500,000 requests / month: 0.00227 USD per each (2.27 USD per 1000)</li> </ul>	<ul style="list-style-type: none"> <li>• SEO improvement [51]</li> <li>• Always updated and most used maps</li> </ul>	<ul style="list-style-type: none"> <li>• No because too expensive</li> </ul>
LocationIQ [53] <a href="#">Demo</a>	<ul style="list-style-type: none"> <li>• Free: 5,000 / day. 2 requests/second</li> <li>• 49\$: 10,000 r/day, 15 r/second</li> </ul>	<ul style="list-style-type: none"> <li>• Limit on requests per second</li> </ul>	<ul style="list-style-type: none"> <li>• No because request per second limitation</li> </ul>
Mapbox [54] - <a href="#">Demo</a>	<ul style="list-style-type: none"> <li>• 1000 sessions (Each session may include unlimited Temporary Geocoding API requests)/ month free</li> <li>• 1,001 – 25,000: 12.5 \$</li> <li>• 25,000 – 100,000: 10 \$</li> <li>• 100,001-500,000: 7.5 \$</li> </ul>	<ul style="list-style-type: none"> <li>• Unlimited calls per session</li> </ul>	<ul style="list-style-type: none"> <li>• No. 2d best price wise</li> </ul>
MapQuest [55] - <a href="#">Demo</a>	<ul style="list-style-type: none"> <li>• 15,000 transactions per month free</li> <li>• 30,000 transactions / month: 119 \$</li> </ul>	<ul style="list-style-type: none"> <li>• Clearly differentiates countries, cities, addresses in search results</li> </ul>	<ul style="list-style-type: none"> <li>• Yes. 1<sup>st</sup> best price wise. No limitations on per second searching</li> </ul>

		<ul style="list-style-type: none"> <li>• Doesn't specify what transaction means</li> </ul>	
TomTom [56]	<ul style="list-style-type: none"> <li>• 50 € per 1000 requests for any end point</li> </ul>	<ul style="list-style-type: none"> <li>• Too expensive</li> </ul>	<ul style="list-style-type: none"> <li>• No, more expensive than Google and worse maps</li> </ul>

## Appendix F – Firestore collections design

### F.1 – Users

One document per user. Each document has an ID. Document content:

- email: <string>
- username <string>
- picturePath <string and reference to path>
- friends <map>: {document ID of friend 1 or reference: <friendship status: string>, document ID of friend 2 or reference: <friendship status: string>, ...}
- trips <list>: [document ID of first trip, document ID of 2d trip, ...]
- notifications <map>: {<notification type>: bool}

The screenshot shows the Firebase Firestore interface. On the left, there's a tree view of collections: 'Root' (with '+ Start collection'), 'trips', and 'users'. Under 'users', a specific document is selected, shown in the center panel with the ID 'Aa6x5je3EsA2q9yPaA4R'. This document has fields: 'email' (string), 'friends' (map), 'notifications' (map), 'picturePath' (null), 'trips' (array), and 'username' (string). The right panel shows the detailed structure of the 'friends' map, which is currently empty. There are also edit and delete icons for each field.

### F.2 - Trips

One document per trip. Each document has an auto-generated unique ID. Document content:

- tripTitle: <string>
- tripOwner: <string: user ID>
- askAllParticipantsDates: <bool>
- finalizedStartDate: <date or null>
- finalizedEndDate: <date or null>
- datesPreferences: <map>: {<user ID>: {1: [<start date>, <end date>], ..}, ..}
- workingDaysAvailability: <map>: {<user ID>: <int>, ..}
- totalDaysAvailability: <map>: {<user ID>: <int>, ..}
- status: <string> upcoming/archived

- creationDatetimeUTC: <datetime>
- lastUpdatedDatetimeUTC: <datetime>
- participantsStatus <map>: {<user ID>: <string (owner, collaborator, ...)>, ..}

The screenshot shows the MongoDB Atlas Data tab interface. At the top, there are tabs for "Data" and "Requests". Below the tabs is a breadcrumb navigation bar: Home > trips > LZnIGdx3i6E0qnFTkeJW. On the right side of the header is a red button labeled "Clear all data". The main area displays a hierarchical document structure under the path "trips > LZnIGdx3i6E0qnFTkeJW". The structure includes fields like "askAllParticipantsDates", "creationDatetimeUTC", "datesPreferences" (with sub-fields for days 0 and 1), "finalizedStartDate", "lastUpdatedDatetimeUTC", "participantsStatus" (with sub-fields for "status" and "totalDaysAvailability"), and "workingDaysAvailability". Each field has edit and delete icons next to it.

Itinerary events are stored as document collection related to the trip and have the following fields:

- event\_category <string>
- title <string>
- start date <datetime>
- end date <datetime>
- address: <string>
- description: <string>
- file\_attachments <list [references to paths or strings]>
- cost: floating-point

- currency: <string>
- split\_type: <string>
- participants: <map>: {<user ID>: <bool>}
- ask\_participants\_if\_they\_will\_join: bool
- status: <string>
- comments: <map>: {<user ID>: {"message": <string>, "datetime": <datetime>}}

Polls are stored as document collection related to the trip and have the following fields:

- owner: <user ID or reference to it>
- title: <string>
- answer\_choices: <list of strings>
- number\_answers\_choosable: <int>
- participants: <list of user ID or reference to it>
- total\_votes\_received: <int>
- total\_votes\_required: <int>
- answers\_received: <map>: {<user ID>: <string>}
- status: <string>: open/closed

## Appendix G – Payment platforms comparison

Platform	Costs
Stripe [65]	<ul style="list-style-type: none"> <li>• standard EU cards: 1.5% + €0.25</li> <li>• 1.9% + €0.25 for premium European Economic Area cards</li> <li>• UK cards: 2.5% + €0.25</li> <li>• US cards: 2.9% + 0.3 USD per transaction</li> <li>• Free tech support and dispute handling</li> </ul>
Square [66]	<ul style="list-style-type: none"> <li>• Optional 29€ / month if you want dispute management and customer support</li> <li>• 2.9% + 0.3 USD per transaction</li> </ul>

## Appendix H – Features available with Pro subscription

Feature	Comments
Sending emails for custom alerts	Without Pro plan, notifications only appear inside the app. This means inviting friends is only available by sharing a link manually.
Saving unlimited attachments to trip items	Not possible without Pro
Export plans as .CSV file	Not possible without Pro
Export plan as a PDF file [67]	Not possible without Pro
Send plan as a PDF file to all attendees	Not possible without Pro
Sync trip to your favorite calendar	Not possible without Pro
Budgeting tools – cost splitting	Not possible without Pro

## Appendix I – UML Diagrams

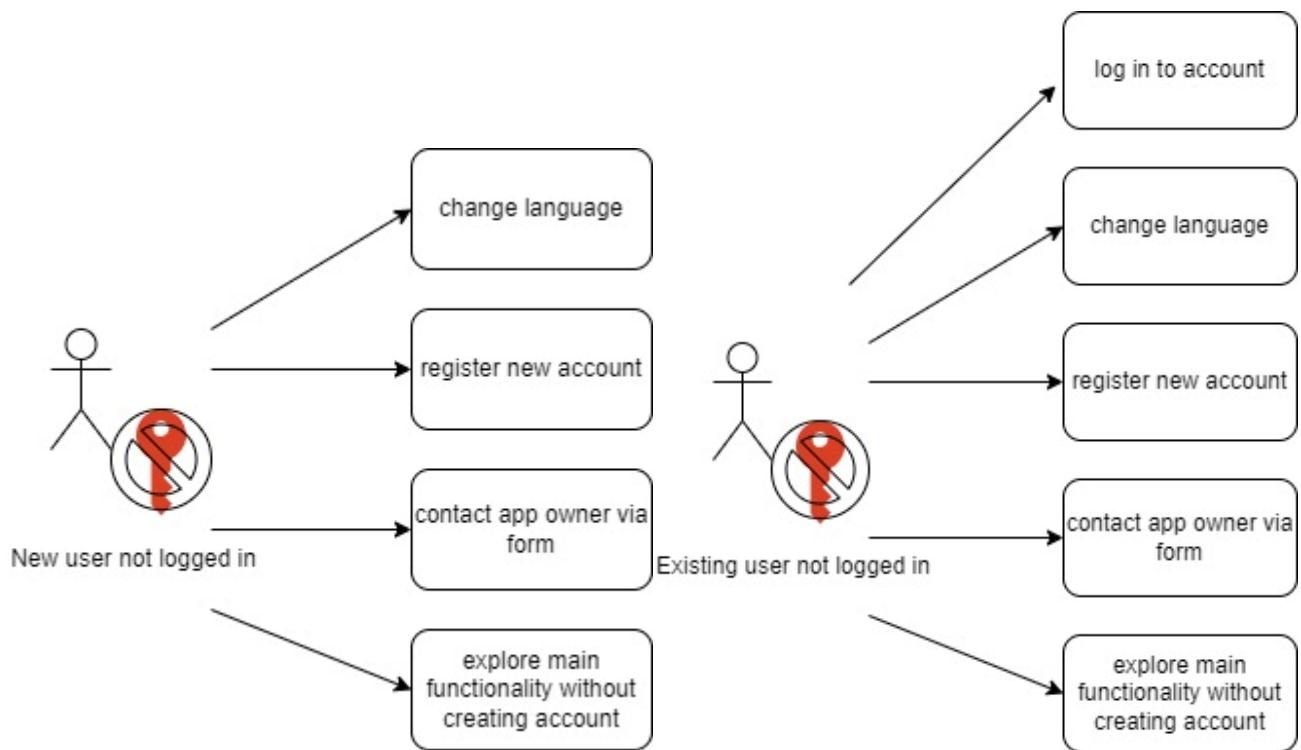


Figure 1 - UML diagram - new unauthenticated user

Figure 2 - UML diagram - unauthenticated user with previously registered account

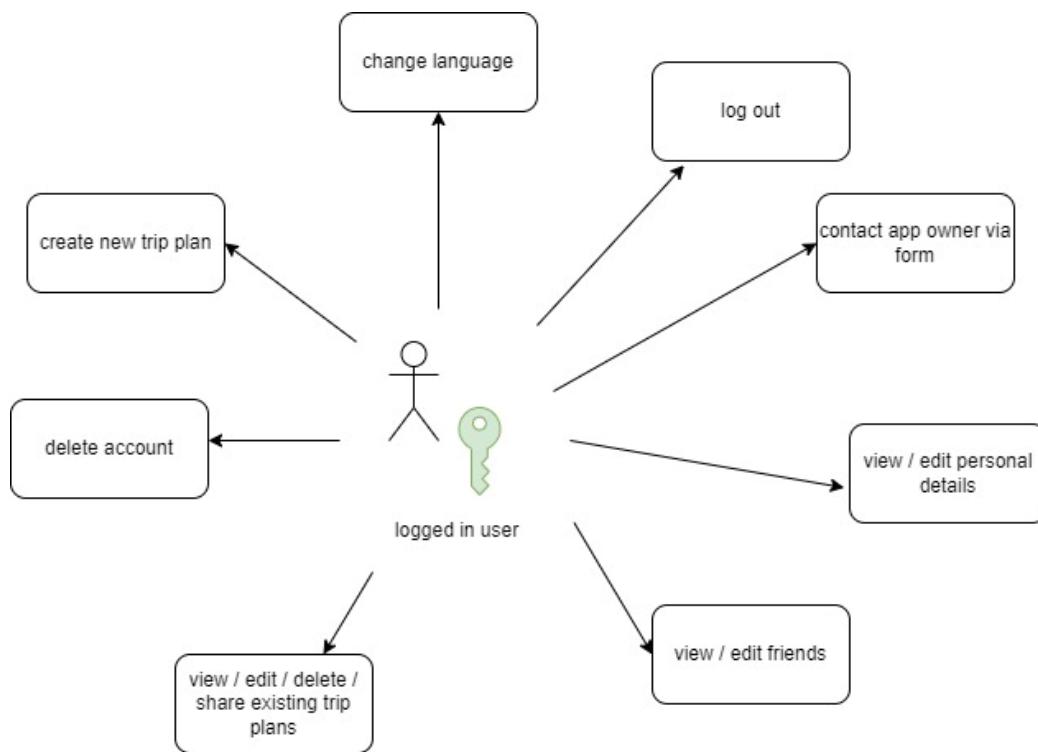


Figure 3 - UML diagram - authenticated users

## Appendix J – Technical infrastructure

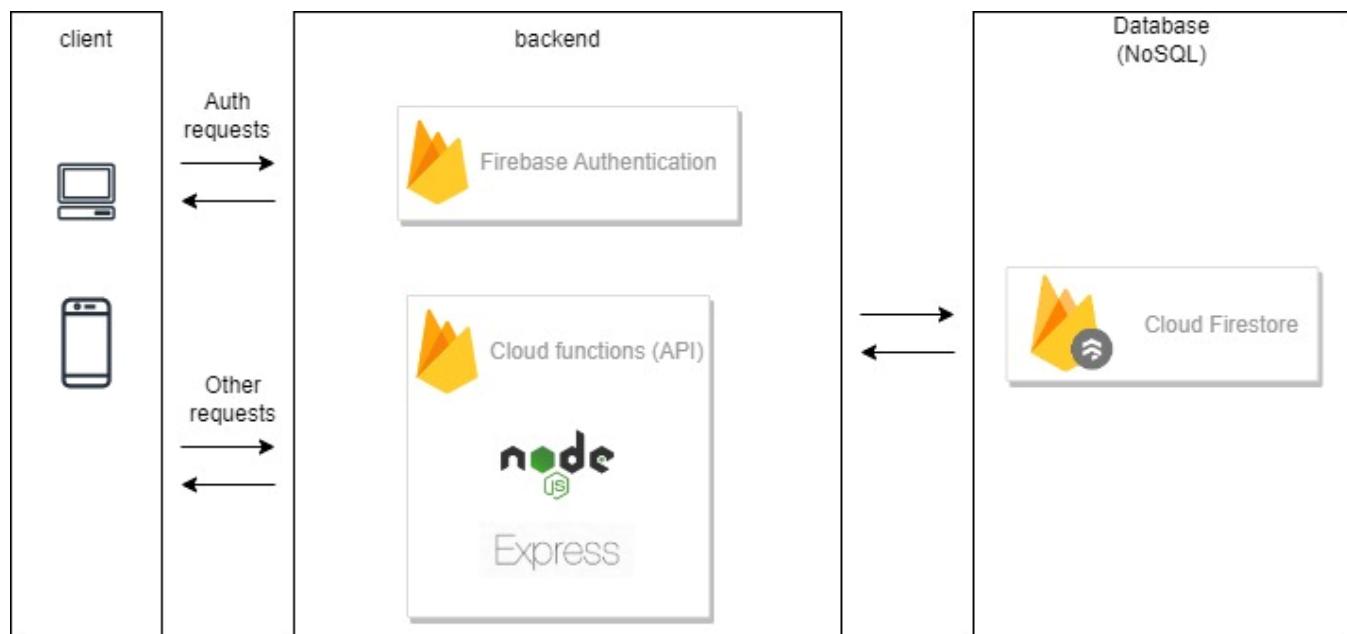


Figure 4 - Technical infrastructure

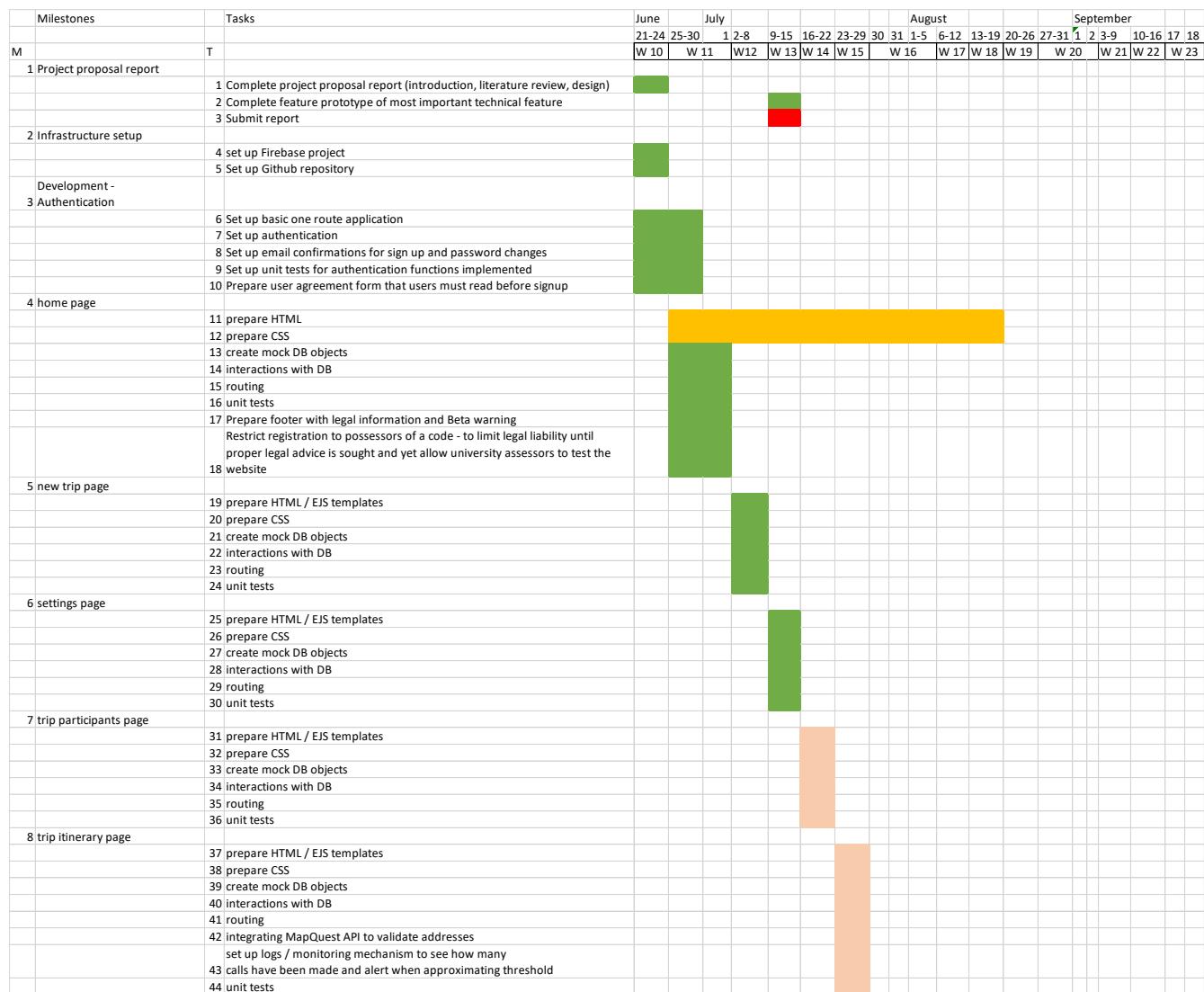
## Appendix K – REST API

Resource	POST	GET	DELETE
/trips	Save new trip details	Retrieve all trips for a user	-
/trips/:id	Updates trip details with new information	Retrieve details of the trip with provided id	Deletes trip with specified ID
/profiles/:id	Save new profile info for user id or update existing profile	Retrieve profile info for user id	Delete profile info for user id
/auth/login	-	Gets template where user can log in	-
/auth/signup	Creates user account	-	-
/auth/sessionLogin	Creates a 8 hours session during which the user is authenticated	-	-
/auth/sessionLogout	Closes user session	-	-
/auth/resetPassword	Sends an email to user with link to reset password	Gets template where user can enter email address and request password reset link	-
/legal/privacy	-	Loads HTML page with privacy conditions	-
/legal/terms	-	Loads HTML page with legal terms	-

/settings	Update a particular notification setting	Gets HTML page displaying current notification settings	-
/	-	Gets template with description of the app	-

## Appendix L – Timelines and Task Breakdown

### L.1 – Gantt Chart



Milestones	Tasks	June	July	August	September															
		21-24	25-30	1 2-8	9-15	16-22	23-29	30	31	1-5	6-12	13-19	20-26	27-31	1	2	3-9	10-16	17	18
M	T	W 10	W 11	W 12	W 13	W 14	W 15		W 16	W 17	W 18	W 19	W 20	W 21	W 22	W 23				
9 trip polls page	44 unit tests 45 prepare HTML / EJS templates 46 prepare CSS 47 create mock DB objects 48 interactions with DB 49 routing 50 unit tests																			
10 trip settings page	51 prepare HTML / EJS templates 52 prepare CSS 53 create mock DB objects 54 interactions with DB 55 routing 56 unit tests 57 Integrating Google SendGrid API for sending emails																			
11 profile personal info page	58 prepare HTML / EJS templates 59 prepare CSS 60 create mock DB objects 61 interactions with DB 62 routing 63 unit tests																			
12 profile friends page	64 prepare HTML / EJS templates 65 prepare CSS 66 create mock DB objects 67 interactions with DB 68 routing 69 unit tests																			
13 security	70 configure Firestore security rules 71 Run security scanners																			
14 integrate Google Adsense	72 Register to Google AdSense 73 Add ads to site																			
15 payment integration	74 Integrate Stripe and register payment history 75 Implement Pro features mechanism																			
16 Final report	76 Write final report 77 Prepare final video presentation 78 Keep progress logs																			
17 Submit final project and report																				

Orange = to do

Red = hard deadlines

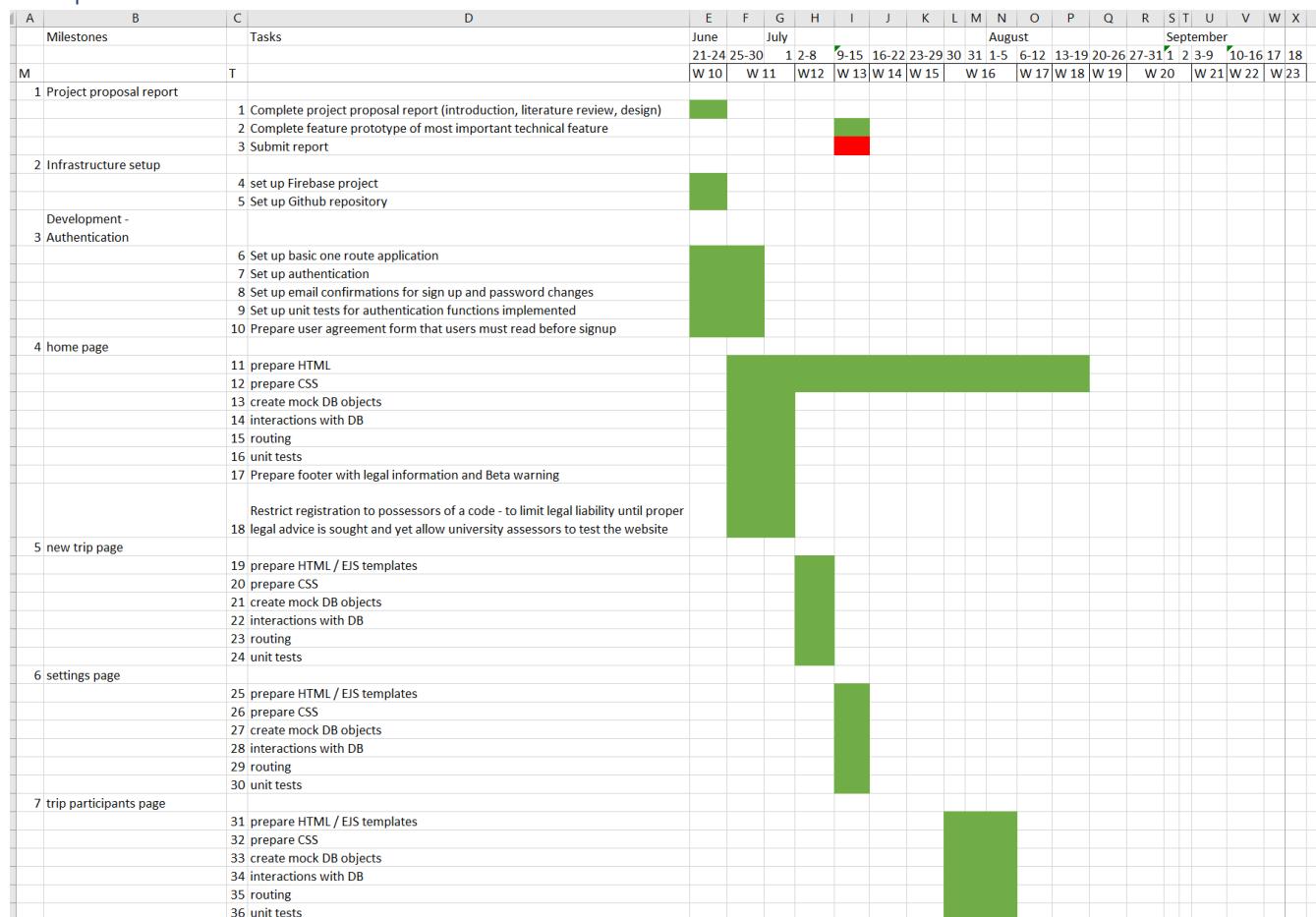
Yellow: in progress

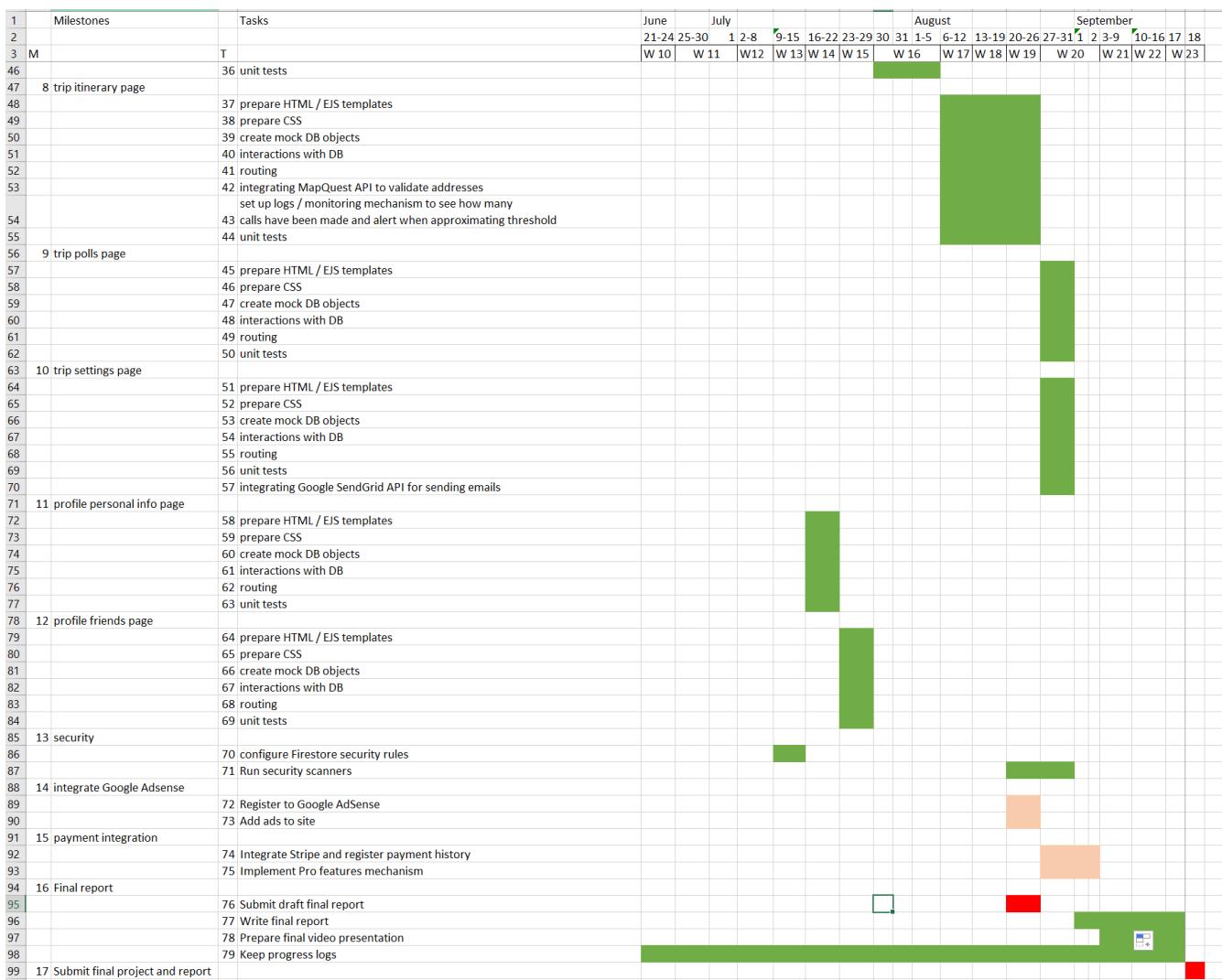
Green = Done

## L.2 – Critical Path Analysis

Milestone	Duration (days)	Dependencies
1	26	
2	4	
3	9	2
4	7	2,3
5	7	2,3
6	7	2,3
7	7	2,3
8	7	2,3
9	7	2,3
10	7	2,3
11	7	2,3
12	7	2,3
13	7	2,3
14	7	2,3
15	7	1-14
16	7	1-14
17	14	1-16
18	2	1-17

## L.3 Updated Gantt chart





## Appendix M – SEO techniques

Technique	Source
Meaningful domain name containing keywords	[73]
<ul style="list-style-type: none"> <li>Choose keywords to optimize (“plan trips as a group”)</li> <li>Repeat keywords at the very top of the page</li> <li>Include keywords at the beginning of titles</li> <li>One keyword per title</li> <li>Name images and alt text with keywords</li> </ul>	[74]
Ensuring all links from my website (backlinks) work	[73]
<ul style="list-style-type: none"> <li>Avoid duplication of content within my website (use content negotiation)</li> <li>Write unique titles, descriptions, content, title tags, meta tags, alt text,</li> <li>Site needs to load as fast as possible, especially on mobile (using PageSpeed Insights [75]) <ul style="list-style-type: none"> <li>Compress images</li> </ul> </li> </ul>	[74]

<ul style="list-style-type: none"> <li>○ Lightweight themes and content</li> <li>● Use <a href="#">Google Search console</a> [76]</li> <li>● Optimize for mobile usability</li> <li>● Each page on the site references all other pages of the site BUT referenced pages must not throw errors (<a href="#">signal not to crawl pages accessible only when logged in</a>)           <ul style="list-style-type: none"> <li>○ Keyword rich anchor text</li> <li>○ <a href="#">Make links crawlable</a></li> </ul> </li> <li>● Maximize time on site (see with <a href="#">Google Analytics</a>)           <ul style="list-style-type: none"> <li>○ Add captivating video at the start</li> <li>○ Make website attractive               <ul style="list-style-type: none"> <li>■ Use infographics</li> <li>■ Animated SVGs</li> </ul> </li> <li>○ Add interactive elements</li> </ul> </li> <li>● Get other sites to link your site           <ul style="list-style-type: none"> <li>○ Advertise on social media</li> <li>○ Make website attractive               <ul style="list-style-type: none"> <li>■ Use infographics</li> <li>■ Animated SVGs</li> </ul> </li> <li>○ Create content hub: review of all similar sites e.g. on an article or Wikipedia</li> <li>○ Create me vs. competitor articles / keywords</li> <li>○ Create linkable assets</li> <li>○ Look for “best sitest that...” articles and pitch product to include it</li> </ul> </li> <li>● Use <a href="https://www.semrush.com/sensor/">https://www.semrush.com/sensor/</a> to see topic ranking</li> <li>● Analyze keywords used by competitors and optimize for those</li> </ul>	<p>[74]</p> <p>[74]</p> <p>[74, 80]</p> <p>[74, 77]</p> <p>[74, 77, 79]</p> <p>[77]</p> <p>[78]</p>
---	---

## Appendix N – Feature prototype unit tests

### N.1 package.json configurations

```
{} package.json > {} scripts
1  {
2    "name": "group_tripper",
3    "version": "1.0.0",
4    "description": "Web app to plan trips as a group",
5    "main": "index.js",
6    "engines": {
7      "node": "18"
8    },
9    "directories": {
10      "doc": "docs"
11    },
12    ▶ Debug
13    "scripts": [
14      "test": "node --test ./test",
15
16    ],
17    "devDependencies": {
18      "webpack": "^5.88.0",
19      "webpack-cli": "^5.1.4",
20      "firebase-functions-test": "^0.2.0",
21      "node-mocks-http": "1.12.2"
22    }
23  },
```

## N.2 – unit tests folders

```
UNIFINALPROJECT
  .firebase
    hosting.cHVibGlj.cache
  docs
  functions
    config
    db
    node_modules
    routers
      authentication.js
      friends.js
      home.js
      legal.js
      notifications.js
      profile.js
      settings.js
      trips.js
  utils
    authUtils.js
    imageSearch.js
    timeUtils.js
  views
  index.js
  package-lock.json
  package.json
  node_modules
  public
  test\functions
    routers
      authentication
      friends
      home
      legal
      notifications
      profile
      settings
      trips
    utils
      authUtils.test.js
      timeUtils.test.js
```

### N.3 – Example unit tests for routes

```
st > functions > routers > home > JS home.test.js > ⚡ describe('homeRouter') callback > ⚡ it("GET / should redirect to /trips when user is logged in") callback
1 import { describe, it } from 'node:test';
2 import assert from 'node:assert';
3 import httpMocks from 'node-mocks-http';
4 import { EventEmitter } from 'events';
5 import { homeRouter } from '../../../../../functions/routers/home.js';
6
7
8 describe('homeRouter', () => {
9     function adminAuth(){
10     }
11
12     it("GET / should redirect to /trips when user is logged in", () => {
13         adminAuth.verifySessionCookie = function(sessionCookie, boolean){
14             return Promise.resolve({
15                 iss: 'example-iss',
16                 name: 'John Doe',
17                 aud: 'example-aud',
18                 auth_time: 1624952023,
19                 user_id: 'example-user-id',
20                 sub: 'example-sub',
21                 iat: 1,
22                 exp: 1,
23                 email: 'john@example.com',
24                 email_verified: true,
25                 firebase: { identities: { email: ['example-email'] }, sign_in_provider: 'password' },
26                 uid: 'example-uid',
27             });
28         };
29
30         // Create a mock request object
31         let request = httpMocks.createRequest({
32             method: 'GET',
33             url: '/',
34             cookies: {
35                 __session: "somesessionstring"
36             }
37         );
38
39         let response = httpMocks.createResponse({eventEmitter: EventEmitter});
40
41         response.on("end", () => {
42             assert.strictEqual(response.statusCode, 302);
43             assert.strictEqual(response.cookies.__session.value, 'somesessionstring');
44             assert.deepEqual(
45                 response.cookies.__session,
46                 { value: 'somesessionstring', options: undefined });
47         );
48
49         let router = homeRouter(
50             adminAuth
51         );
52         router.handle(request, response);
53     });
54 })
```

#### N.4 Unit tests results

```
C:\Users\ventafri\Desktop\Uni\year 3\UniFinalProject>npm test

> group_tripper@1.0.0 test
> node --test ./test

[] authenticationRouter
  [] GET /login should render authentication.ejs with the correct payload (4.0977ms)
[] authenticationRouter (5.2312ms)

[] authenticationRouter
  [] GET /resetPassword should render authentication.ejs with the correct payload (4.3782ms)
  [] POST /resetPassword called with valid email should send email and render authentication.ejs with the correct payload (1.1323ms)
  [] POST /resetPassword called with invalid email should render authentication.ejs with the error message (0.9784ms)
  [] POST /resetPassword called with invalid email but auth/user-not-found error should render authentication.ejs with the success message (0.8242ms)
[] authenticationRouter (8.5117ms)

[] POST /:tripId/polls/new should redirect to /auth/login when user is logged out (0.4727ms)
[] POST /:tripId/polls/new should send an error if anything goes wrong (0.4405ms)
[] DELETE /:tripId/polls/:pollID should return status 200 when user is logged in (0.5665ms)
[] DELETE /:tripId/polls/:pollID should redirect to /auth/login when user is logged out (0.3758ms)
[] DELETE /:tripId/polls/:pollID should send an error if anything goes wrong (0.6715ms)
[] POST /:tripId/polls/:pollID should return status 200 when user is logged in (0.5251ms)
[] POST /:tripId/polls/:pollID should redirect to /auth/login when user is logged out (0.4995ms)
[] POST /:tripId/polls/:pollID should send an error if anything goes wrong (0.3271ms)
[] GET /:tripId/polls/:pollID should render tripPolls.ejs with details when user is logged in (1.7809ms)
[] GET /:tripId/polls/:pollID should redirect to /trips when user is logged out (0.2735ms)
[] GET /:tripId/polls/:pollID should send an error if anything goes wrong (1.696ms)
[] tripsRouterTripPolls (19.3006ms)

[] tripsRouterTripSettings
  [] GET /:id/settings should render tripSettings.ejs with details when user is logged in (5.9405ms)
  [] GET /:id/settings should redirect to /trips when user is logged out (0.9801ms)
  [] GET /:id/settings should send an error if anything goes wrong (0.5809ms)
[] tripsRouterTripSettings (8.8672ms)

[] attachCsrfToken
  [] should set the cookie value when the request URL matches (0.3424ms)
  [] should not set the cookie value when the request URL does not match (0.0959ms)
[] attachCsrfToken (1.5373ms)

[] getUserSessionDetails
  [] should return authenticated user session details when session cookie is valid (0.7665ms)
  [] should return null user session details when user is not authenticated (0.8292ms)
  [] should return null user session details and an error when there is an error (0.2938ms)
[] getUserSessionDetails (2.014ms)

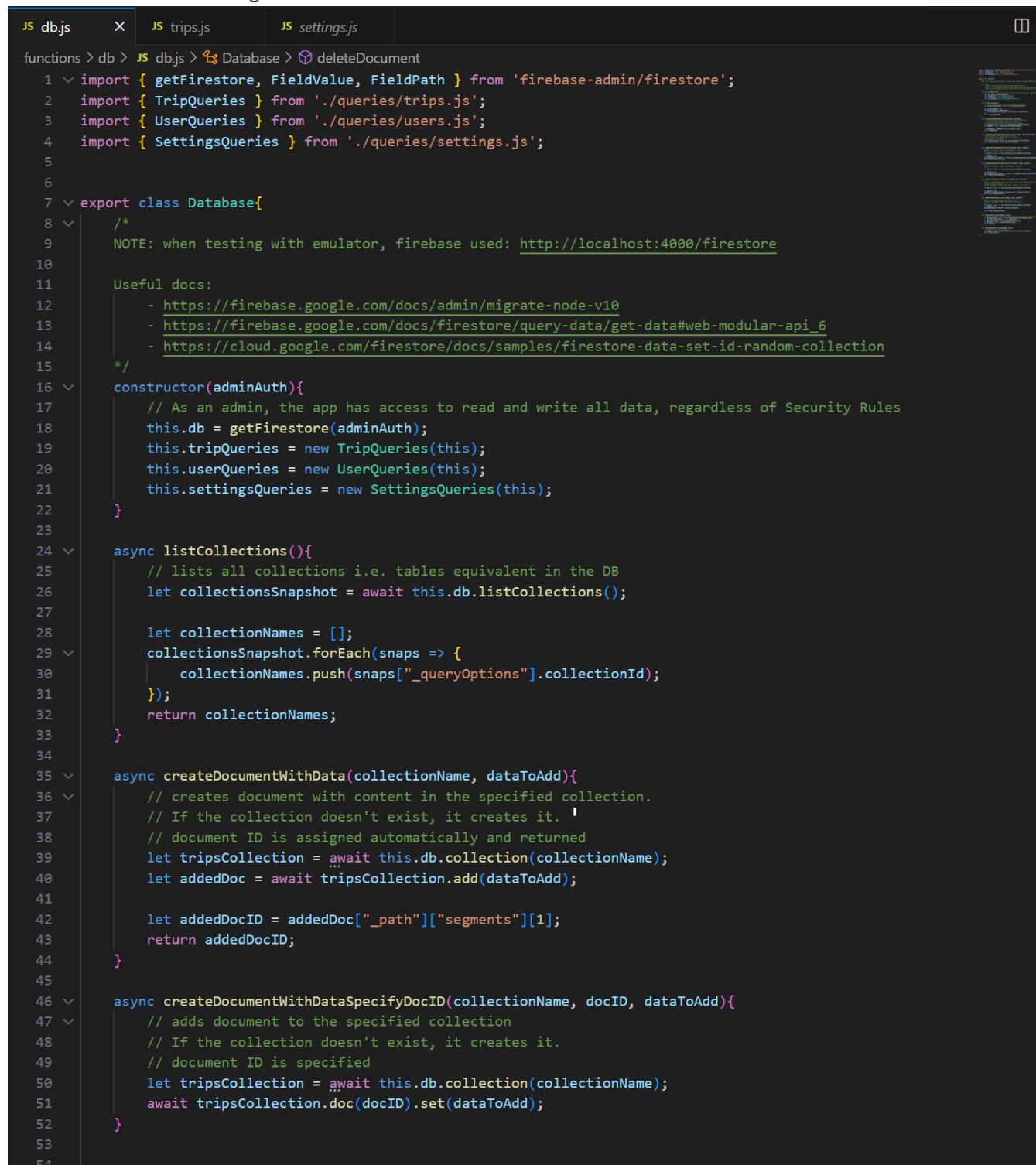
[] commonDateRanges
  [] should return common dates preferences between 2 users (3.7768ms)
  [] should return 2 common dates preferences between 3 users (1.543ms)
  [] should return no common dates preferences (0.3498ms)
[] commonDateRanges (7.0973ms)

[] tests 23
[] pass 23
[] fail 0
[] cancelled 0
[] skipped 0
[] todo 0
[] duration_ms 6717.2899

C:\Users\ventafri\Desktop\Uni\year 3\UniFinalProject>
```

## Appendix O – Feature prototype database interactions

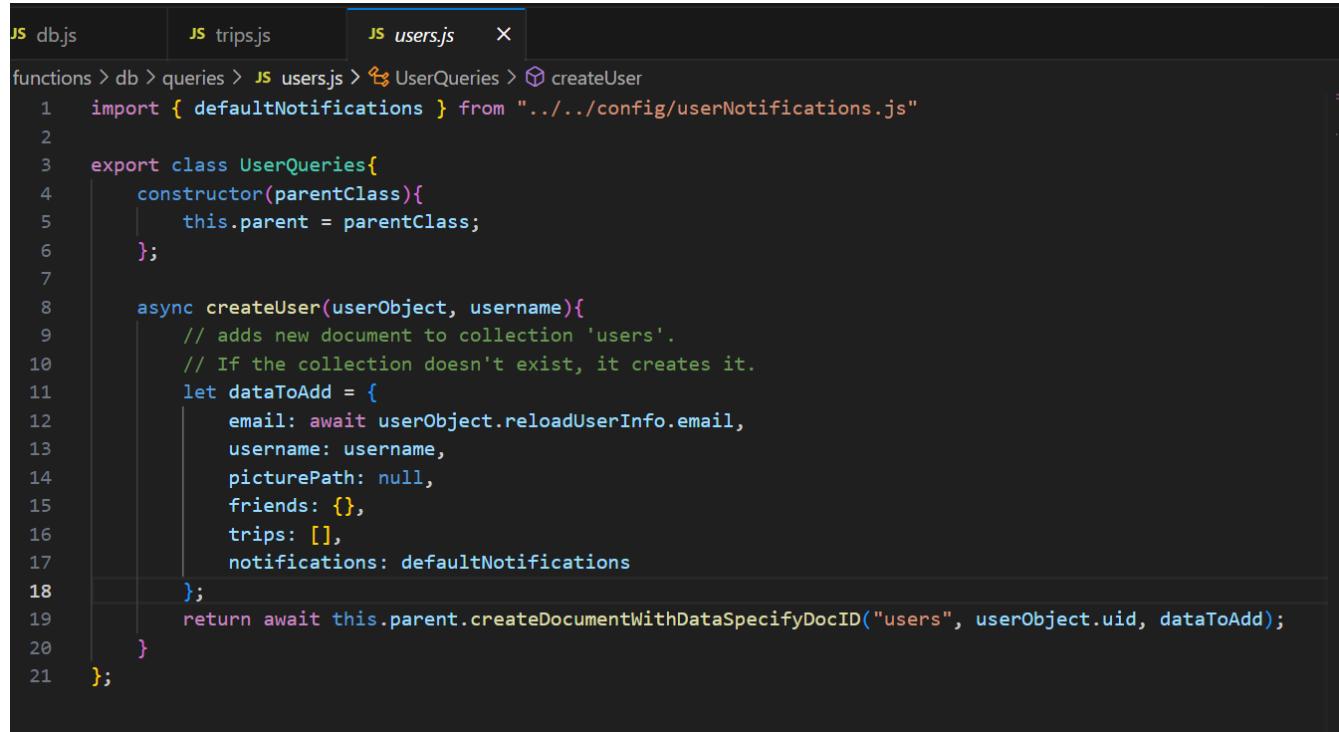
### O.1 – DB class: contains generic methods to interact with Firestore



The screenshot shows a code editor window with the file `db.js` open. The code defines a `Database` class with various methods for interacting with Firestore. The code includes imports for `getFirestore`, `FieldValue`, and `FieldPath` from `firebase-admin/firestore`, and for `TripQueries`, `UserQueries`, and `SettingsQueries` from `./queries/trips.js`, `./queries/users.js`, and `./queries/settings.js` respectively. The `Database` class constructor takes `adminAuth` as a parameter and initializes `this.db`, `this.tripQueries`, `this.userQueries`, and `this.settingsQueries`. It also provides a `listCollections` method to list all collections in the database and a `createDocumentWithData` method to create a document in a specified collection. A note at the top of the class specifies that when testing with an emulator, Firebase uses `http://localhost:4000.firebaseio.json`.

```
JS db.js X JS trips.js JS settings.js
functions > db > JS db.js > 📁 Database > ⚙️ deleteDocument
1 1 import { getFirestore, FieldValue, FieldPath } from 'firebase-admin/firestore';
2 2 import { TripQueries } from './queries/trips.js';
3 3 import { UserQueries } from './queries/users.js';
4 4 import { SettingsQueries } from './queries/settings.js';
5
6
7 7 export class Database{
8 8 /*
9 9   NOTE: when testing with emulator, firebase used: http://localhost:4000.firebaseio.json
10
11 11   Useful docs:
12 12     - https://firebase.google.com/docs/admin/migrate-node-v10
13 13     - https://firebase.google.com/docs/firestore/query-data/get-data#web-modular-api_6
14 14     - https://cloud.google.com/firestore/docs/samples/firestore-data-set-id-random-collection
15 15 */
16 16 constructor(adminAuth){
17 17   // As an admin, the app has access to read and write all data, regardless of Security Rules
18 18   this.db = getFirestore(adminAuth);
19 19   this.tripQueries = new TripQueries(this);
20 20   this.userQueries = new UserQueries(this);
21 21   this.settingsQueries = new SettingsQueries(this);
22 22 }
23
24 24 async listCollections(){
25 25   // lists all collections i.e. tables equivalent in the DB
26 26   let collectionsSnapshot = await this.db.listCollections();
27
28 28   let collectionNames = [];
29 29   collectionsSnapshot.forEach(snaps => {
30 30     collectionNames.push(snaps[_queryOptions].collectionId);
31 31   });
32 32   return collectionNames;
33 33 }
34
35 35 async createDocumentWithData(collectionName, dataToAdd){
36 36   // creates document with content in the specified collection.
37 37   // If the collection doesn't exist, it creates it.
38 38   // document ID is assigned automatically and returned
39 39   let tripsCollection = await this.db.collection(collectionName);
40 40   let addedDoc = await tripsCollection.add(dataToAdd);
41
42 42   let addedDocID = addedDoc[_path][segments][1];
43 43   return addedDocID;
44 44 }
45
46 46 async createDocumentWithDataSpecifyDocID(collectionName, docID, dataToAdd){
47 47   // adds document to the specified collection
48 48   // If the collection doesn't exist, it creates it.
49 49   // document ID is specified
50 50   let tripsCollection = await this.db.collection(collectionName);
51 51   await tripsCollection.doc(docID).set(dataToAdd);
52 52 }
53
54 54
```

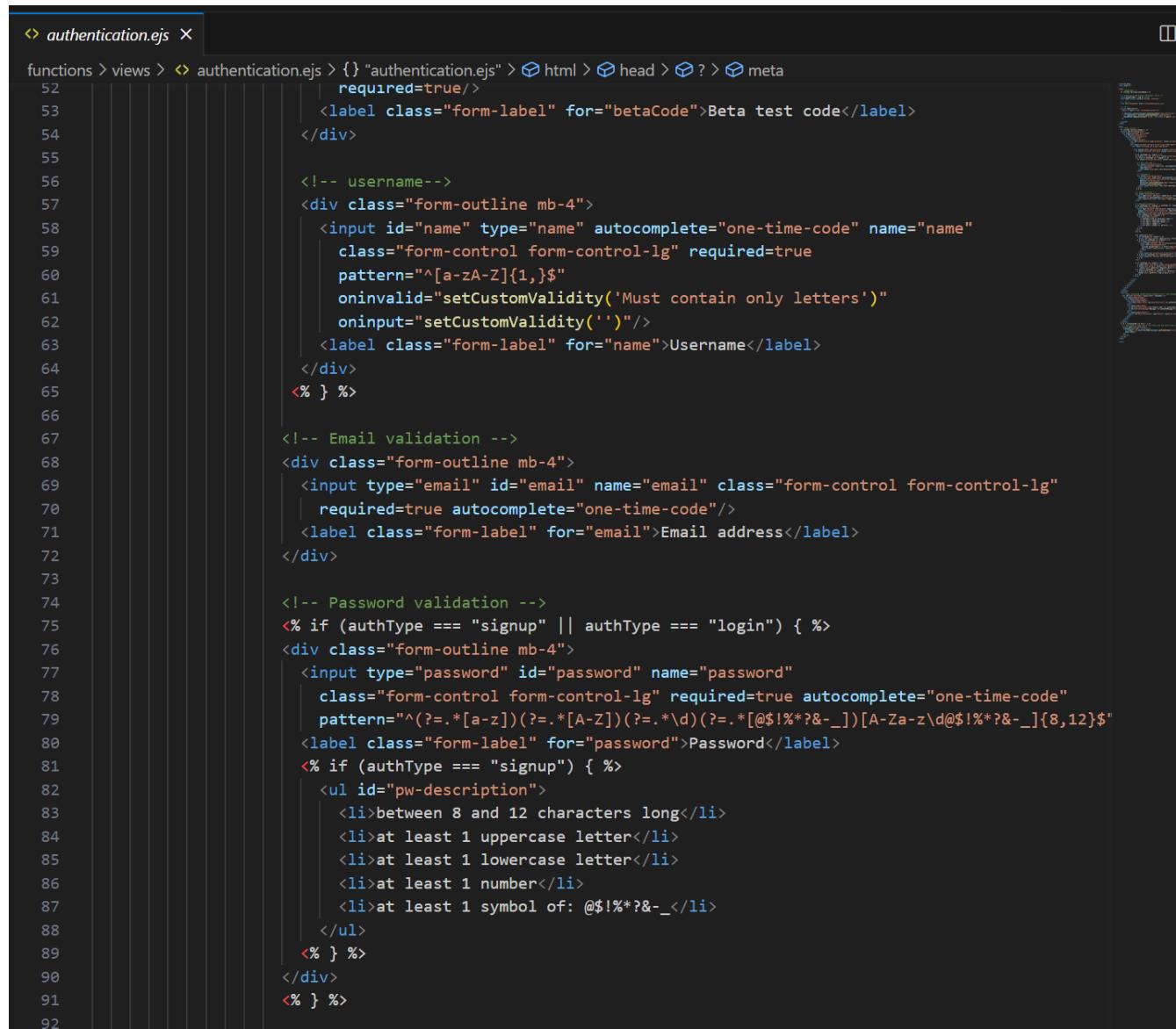
## O.2 – Sample queries



```
JS db.js JS trips.js JS users.js X
functions > db > queries > JS users.js > UserQueries > createUser
1 import { defaultNotifications } from "../../config/userNotifications.js"
2
3 export class UserQueries{
4     constructor(parentClass){
5         this.parent = parentClass;
6     }
7
8     async createUser(userObject, username){
9         // adds new document to collection 'users'.
10        // If the collection doesn't exist, it creates it.
11        let dataToAdd = {
12            email: await userObject.reloadUserInfo.email,
13            username: username,
14            picturePath: null,
15            friends: {},
16            trips: [],
17            notifications: defaultNotifications
18        };
19        return await this.parent.createDocumentWithDataSpecifyDocID("users", userObject.uid, dataToAdd);
20    }
21}
```

## Appendix P – Feature prototype input validation and sanitization

### P.1 - Front-end validation



```
functions > views > authentication.ejs > {} "authentication.ejs" > html > head > ? > meta
52 |     required=true/>
53 |         <label class="form-label" for="betaCode">Beta test code</label>
54 |     </div>
55 |
56     <!-- username-->
57     <div class="form-outline mb-4">
58         <input id="name" type="name" autocomplete="one-time-code" name="name"
59             class="form-control form-control-lg" required=true
60             pattern="^[a-zA-Z]{1,}$"
61             oninvalid="setCustomValidity('Must contain only letters')"
62             oninput="setCustomValidity('')"/>
63         <label class="form-label" for="name">Username</label>
64     </div>
65     <% } %>
66
67     <!-- Email validation -->
68     <div class="form-outline mb-4">
69         <input type="email" id="email" name="email" class="form-control form-control-lg"
70             required=true autocomplete="one-time-code"/>
71         <label class="form-label" for="email">Email address</label>
72     </div>
73
74     <!-- Password validation -->
75     <% if (authType === "signup" || authType === "login") { %>
76     <div class="form-outline mb-4">
77         <input type="password" id="password" name="password"
78             class="form-control form-control-lg" required=true autocomplete="one-time-code"
79             pattern="^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[$!%*?&-_])[A-Za-z\d$!%*?&-_]{8,12}$">
80         <label class="form-label" for="password">Password</label>
81         <% if (authType === "signup") { %>
82             <ul id="pw-description">
83                 <li>between 8 and 12 characters long</li>
84                 <li>at least 1 uppercase letter</li>
85                 <li>at least 1 lowercase letter</li>
86                 <li>at least 1 number</li>
87                 <li>at least 1 symbol of: @$!%*?&-_-</li>
88             </ul>
89             <% } %>
90         </div>
91     <% } %>
```

## P.2 – Backend sanitization

The screenshot shows the Visual Studio Code interface with the following details:

- Explorer View:** Shows the project structure. The `authentication.js` file is currently selected.
- Editor View:** Displays the `authentication.js` file content. The code handles user sign-up logic, including validation rules for name, email, and password, and handles terms and conditions acceptance.
- Timeline View:** Located at the bottom left, it shows the history of changes made to the file.

```
functions > routers > authentication.js > authenticationRouter
  ...
  ...
  router.post(
    "/signup", [
      check("name").trim().escape().isLength({ min: 1 }).withMessage('Username must contain at least one character'),
      check("email").isEmail().withMessage('Email address is invalid'),
      // https://express-validator.github.io/docs/api/validation-chain/#isStrongPassword
      check("password").isStrongPassword({
        minLength: 8,
        maxLength: 12,
        minLowercase: 1,
        minUppercase: 1,
        minNumbers: 1,
        minSymbols: 1,
        returnScore: false,
        pointsPerUnique: 1,
        pointsPerRepeat: 0.5,
        pointsForContainingLower: 10,
        pointsForContainingUpper: 10,
        pointsForContainingNumber: 10,
        pointsForContainingSymbol: 10,
      }).withMessage('Password must be between 8 and 12 characters long, contain at least 1 uppercase letter, at least 1 lowercase letter, at least 1 number, and at least one symbol'),
      check("termsandconditions").trim().escape().isLength({ min: 2 }).withMessage('Terms and conditions must be accepted in order to register')
    ],
    (request, response) => {
      let authTemplate = path.join(__dirname, '../views/authentication.ejs');

      try {
        // throw error if anything fails in the validation
        const result = validationResult(request);
        result.throw();
        // accessing request.body.<attribute> now returns sanitized input as specified above
      } catch (e) {
        // reload the signup page, which will display a modal with error message
        let payload = {authType: "signup", statusCode: 400,
                      authInfoMessage: e.array({ onlyFirstError: true })[0].msg,
                      authInfoTitle: authInfoErrorTitle};

        return response.status(400).render(authTemplate, payload);
        throw new Error('breaking');
      }

      if (request.body.termsandconditions !== "on") {
        // reload the signup page, which will display a modal with error message
        let payload = {authType: "signup", statusCode: 400, authInfoTitle: authInfoErrorTitle,
                      authInfoMessage: "Terms and conditions must be accepted in order to register"};
        response.status(400).render(authTemplate, payload);
        throw new Error('breaking');
      }

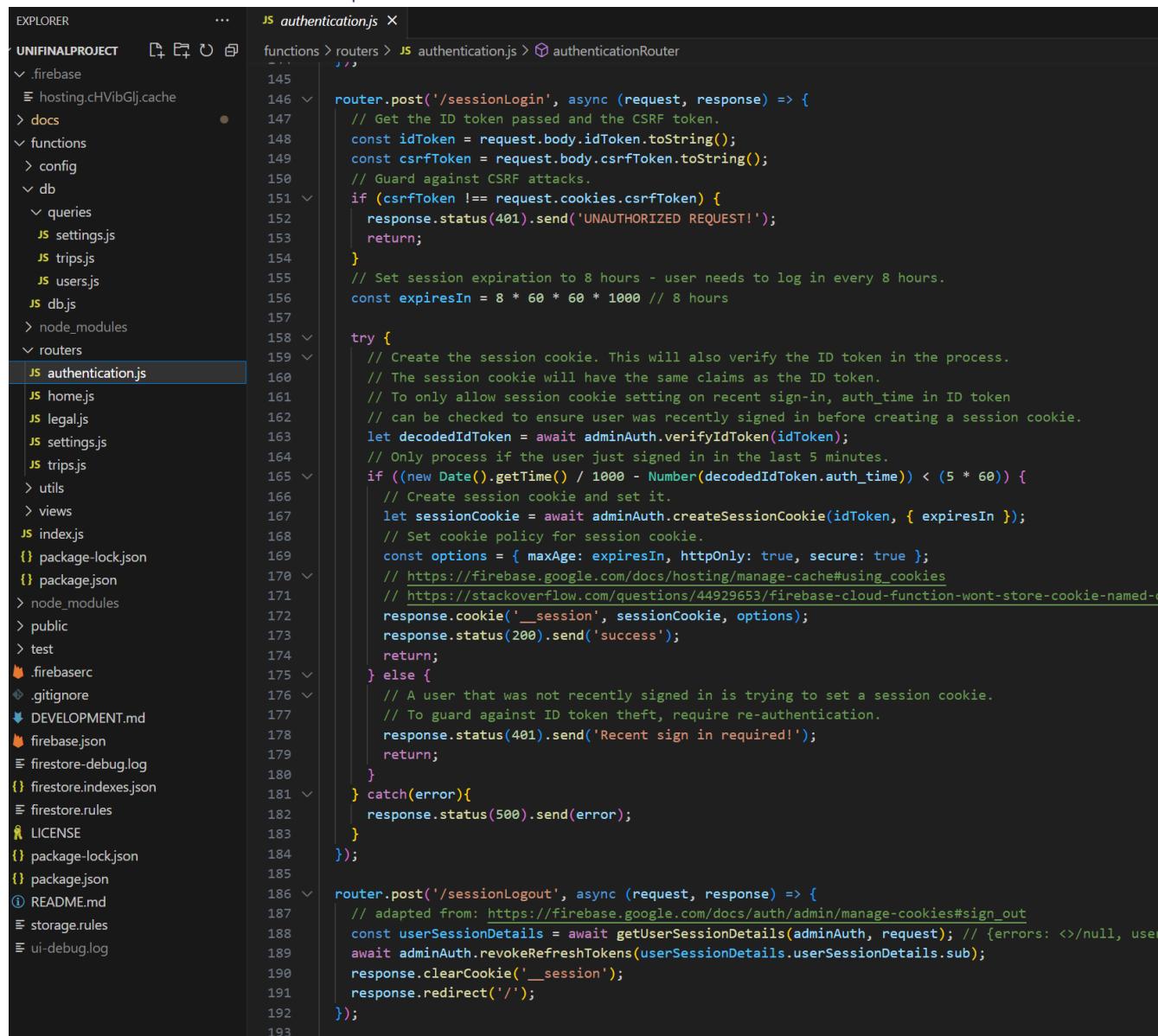
      if (request.body.betaCode !== "girotondo") {
        // reload the signup page, which will display a modal with error message
        let payload = {authType: "signup", statusCode: 400, authInfoTitle: authInfoErrorTitle,
                      authInfoMessage: "Please be patient.. For the time being, registration is restricted"};
        response.status(400).render(authTemplate, payload);
        throw new Error('breaking');
      }

      // create user in database
      createUserWithEmailAndPassword(clientAuth, request.body.email, request.body.password)
        .then((userCredential) => {
          // If the new account was created, the user is signed in automatically.
          const user = userCredential.user;

          updateProfile(user, {
            displayName: request.body.name
          });
        });
    }
  );
}
```

## Appendix Q – Authentication

### Q.1 – session creation API end point



```
functions > routers > JS authentication.js > authenticationRouter
  ...
  145
  146 router.post('/sessionLogin', async (request, response) => {
  147   // Get the ID token passed and the CSRF token.
  148   const idToken = request.body.idToken.toString();
  149   const csrfToken = request.body.csrfToken.toString();
  150   // Guard against CSRF attacks.
  151   if (csrfToken !== request.cookies.csrfToken) {
  152     response.status(401).send('UNAUTHORIZED REQUEST!');
  153     return;
  154   }
  155   // Set session expiration to 8 hours - user needs to log in every 8 hours.
  156   const expiresIn = 8 * 60 * 60 * 1000 // 8 hours
  157
  158   try {
  159     // Create the session cookie. This will also verify the ID token in the process.
  160     // The session cookie will have the same claims as the ID token.
  161     // To only allow session cookie setting on recent sign-in, auth_time in ID token
  162     // can be checked to ensure user was recently signed in before creating a session cookie.
  163     let decodedIdToken = await adminAuth.verifyIdToken(idToken);
  164     // Only process if the user just signed in in the last 5 minutes.
  165     if ((new Date().getTime() / 1000 - Number(decodedIdToken.auth_time)) < (5 * 60)) {
  166       // Create session cookie and set it.
  167       let sessionCookie = await adminAuth.createSessionCookie(idToken, { expiresIn });
  168       // Set cookie policy for session cookie.
  169       const options = { maxAge: expiresIn, httpOnly: true, secure: true };
  170       // https://firebase.google.com/docs/hosting/manage-cache#using_cookies
  171       // https://stackoverflow.com/questions/44929653/firebase-cloud-function-wont-store-cookie-named-cookies
  172       response.cookie('__session', sessionCookie, options);
  173       response.status(200).send('success');
  174       return;
  175     } else {
  176       // A user that was not recently signed in is trying to set a session cookie.
  177       // To guard against ID token theft, require re-authentication.
  178       response.status(401).send('Recent sign in required!');
  179       return;
  180     }
  181   } catch(error){
  182     response.status(500).send(error);
  183   }
  184 });
  185
  186 router.post('/sessionLogout', async (request, response) => {
  187   // adapted from: https://firebase.google.com/docs/auth/admin/manage-cookies#sign_out
  188   const userSessionDetails = await getUserSessionDetails(adminAuth, request); // {errors: <>/null, userSessionDetails}
  189   await adminAuth.revokeRefreshTokens(userSessionDetails.userSessionDetails.sub);
  190   response.clearCookie('__session');
  191   response.redirect('/');
  192 });
  193
```

## Q.2 – Frontend Login call

```

EXPLORER ... JS authentication.js X
UNFINALPROJECT ...
  .firebase
    hosting.cHVibGlj.cache
  > docs
  > functions
    > config
    > db
      > queries
        JS settings.js
        JS trips.js
        JS users.js
        JS db.js
      > node_modules
    > routers
      JS authentication.js
      JS home.js
      JS legal.js
      JS settings.js
      JS trips.js
    > utils
    > views
      JS index.js
    {} package-lock.json
    {} package.json
  > node_modules
  > public
    > assets
    > css
  > js
    JS authentication.js
    JS firebaseConfig.js
    JS settings.js
    JS trips.js
    ★ favicon.ico
  > test
    .firebaserc
    .gitignore
    DEVELOPMENT.md
    firebase.json
    firestore-debug.log
    firestore.indexes.json
    firestore.rules
    LICENSE
    package-lock.json
    package.json
    README.md
    storage.rules
    ui-debug.log

JS authentication.js
public > js > JS authentication.js > ...
  ...
  33  };
  34
  35  export function logIn() {
  36    const email = $("#email").val();
  37    const password = $("#password").val();
  38
  39    // https://firebase.google.com/docs/auth/admin/manage-cookies
  40    // As httpOnly cookies are to be used, do not persist any state client side.
  41    // https://firebase.google.com/docs/auth/web/auth-state-persistence#web-modular-api_1
  42    setPersistence(auth, inMemoryPersistence)
  43    .then(() => {
  44      /* Called when user confirms login */
  45      return signInWithEmailAndPassword(auth, email, password);
  46    })
  47    .then(userCredential => {
  48      // Signed in
  49      const user = userCredential.user;
  50      if (!user.emailVerified === true) {
  51        signOut(auth).then(() => {
  52          showLoginPageModal("...Almost there!",
  53                             "Please verify your email before logging in");
  54        })
  55      } else {
  56        // Get the user's ID token as it is needed to exchange for a session cookie.
  57        return auth.currentUser.getIdToken().then(function(idToken) {
  58          // Session login endpoint is queried and the session cookie is set.
  59          // CSRF protection should be taken into account.
  60          const csrfToken = getCookie('csrfToken');
  61
  62          return postIdTokenToSessionLogin('/auth/sessionLogin', idToken, csrfToken);
  63        });
  64      }
  65    })
  66    .then(() => {
  67      // A page redirect would suffice as the persistence is set to NONE.
  68      return signOut(auth);
  69    })
  70    .then(() => {
  71      // redirect home
  72      window.location.assign('/');
  73    })
  74    .catch(error => {
  75      const statusCode = error.code;
  76      const authInfoMessage = error.message;
  77      showLoginPageModal("Ops! Looks like something went wrong",
  78                         statusCode + "\n" + authInfoMessage);
  79    });
  80  }
  81

```

## Appendix R – GitHub repository and Git usage

### R.1 - git log

```
C:\WINDOWS\system32\cmd.exe - git log
commit 7a2cd8bb1f2dcf5be1ca56b9ba86efffea0bcc92f (HEAD -> main, origin/main, origin/HEAD)
Author: Susanna Ventafridda <susanna.ventafridda@gmail.com>
Date: Sun Jul 16 18:02:14 2023 +0200

    Improved Db comments

commit 4bb5dc93e07409cd79c08c190a05254ab2a85229
Author: Susanna Ventafridda <susanna.ventafridda@gmail.com>
Date: Sat Jul 15 23:51:07 2023 +0200

    Fixed bug UID

commit 5df823de4ac7438aef948e39b0b573aeb735e603
Author: Susanna Ventafridda <susanna.ventafridda@gmail.com>
Date: Sat Jul 15 23:05:17 2023 +0200

    Removed print statements

commit 3e530d9299cf212352dbd33272588ca8e9aa2630
Author: Susanna Ventafridda <susanna.ventafridda@gmail.com>
Date: Sat Jul 15 23:01:46 2023 +0200

    Added needed beta test code to register

commit 1c908bd0d57982be676a558b76788c4426b544bf
Author: Susanna Ventafridda <susanna.ventafridda@gmail.com>
Date: Sat Jul 15 22:40:43 2023 +0200

    Implemented unit tests settings router

commit 5e3dde9d1e7e83c0140e2c070fc20c487ccb46f1
Author: Susanna Ventafridda <susanna.ventafridda@gmail.com>
Date: Sat Jul 15 21:33:51 2023 +0200

    Functioning toggles save settings preferences to DB

commit ab9bf08b10fabbb708fa58e7a571657a82efc5dfe
Author: Susanna Ventafridda <susanna.ventafridda@gmail.com>
Date: Sat Jul 15 20:38:18 2023 +0200

    Added settings router - get current user notifications settings

commit 57a14e126e8ce1ee121009cc8c8853988069ffaa
Author: Susanna Ventafridda <susanna.ventafridda@gmail.com>
Date: Sat Jul 15 18:17:11 2023 +0200

    Implemented unit tests for all routes in trips router

commit cc5a6cf7cf218e1078035a8e43a6c385ec337b8
Author: Susanna Ventafridda <susanna.ventafridda@gmail.com>
Date: Sat Jul 15 16:23:50 2023 +0200

    Fixed existing unit tests

commit eaa278ebdae0d19a1840c3e4a3c72b9a1c4bf0cc
Author: Susanna Ventafridda <susanna.ventafridda@gmail.com>
Date: Sat Jul 15 15:06:01 2023 +0200

    Adjusted phone view trips page

commit 54669de2d5e9b38c49321cbaf45ff249e0f5f91
Author: Susanna Ventafridda <susanna.ventafridda@gmail.com>
Date: Sat Jul 15 14:54:32 2023 +0200

    Implemented Update, Delete DB operations and made trips page fully integrated with DB actions vs previous mock objects

commit 58a2d189e4163cdf0300d735af17b523b865c57b
Author: Susanna Ventafridda <susanna.ventafridda@gmail.com>
Date: Mon Jul 10 22:18:58 2023 +0200

    Implemented correct fields in database when a trip is saved and added users collection when users signs up. WIP
    updating existing fields in document

commit 17c8fc16d89f6ffc8f19c39766234256d0fd17f9
Author: Susanna Ventafridda <susanna.ventafridda@gmail.com>
Date: Sun Jul 9 20:07:14 2023 +0200

    Added documentation

commit 06f4316388aeb33ff867fc4bc55731069b2ae76d
Author: Susanna Ventafridda <susanna.ventafridda@gmail.com>
Date: Sun Jul 9 20:02:06 2023 +0200
}
```

## R.2 – GitHub repository

The screenshot shows a GitHub repository page for 'UniFinalProject'. The repository is private and has 1 unwatcher, 0 forks, and 0 stars. It contains 2 branches and 0 tags. The main branch is not protected. There are 89 commits from user '7a2cd8b' made 22 minutes ago. The commits include improvements to Db comments, public toggles, and index.html pages, as well as initial commits and fixes for various files like .firebaserc, .gitignore, and package-lock.json. The repository has an Apache-2.0 license and 1 watcher. It also includes sections for Releases, Packages, Languages (JavaScript 55.6%, EJS 39.3%, CSS 5.1%), and Suggested Workflows for Actions Importer, Webpack, and Gulp.

## Appendix S – Blackbox testing

n.	Feature	Testing procedure	Test result
1	Experience main functionality without logging in	When users are not logged in, although they can't directly interact with the actual functionality of the website, they can still see all functionality available and how the website looks like from the home page. Here, a series of gifs	Pass

		demonstrate the typical user flow when organizing a trip.	
2	User registration	Users can reach the signup page either by clicking on the call for action button visible on the hero image from the home page or by opening the burger menu on the navigation bar (top-right) and clicking on 'Sign up'. Users then need to enter a code ('girotondo') in order to sign up, fill in their username, email address, and password, accept terms and conditions, and then click register . Check emails and click on registration confirmation link.	Pass
3	User login	Open burger menu from navigation bar. Click 'Log in'. Enter email and password and then click 'Log in'.	Pass
4	Password reset	Open burger menu from navigation bar. Click 'Log in'. Enter email and password and then click 'Log in'. Click on 'forgot password?' link. Follow instructions received via email and log in again.	Pass
5	Display clear user agreement and data privacy documentation	Privacy policy and terms and conditions must be accepted when signing up and are always available in the footer of each page.	Pass
6	Logout	After logging in, can click on navbar burger menu, 'Log out'. The action redirects home. If I try to access pages that require authentication at this point ( <a href="https://grouptripper-3c7f1.web.app/trips">https://grouptripper-3c7f1.web.app/trips</a> ), I am redirected to <a href="https://grouptripper-3c7f1.web.app/auth/login">https://grouptripper-3c7f1.web.app/auth/login</a>	Pass
7	Delete account	As authenticated user, I can click on the burger menu from the navbar and then on 'Profile'. I can click on 'Delete Profile'. A popup alert asks me to confirm and if I click OK, my profile is	Pass

		deleted from the database, I am logged out and redirected home. All trips for which I am owner are deleted. Notifications I have sent are removed from my friends' profiles.	
8	Communicating to developers / product owners	On the footer of each page I can see the email address of GroupTripper: contact.grouptripper@gmail.com	Pass
9	Create a new plan	I am logged in. From the home page ( <a href="https://grouptripper-3c7f1.web.app/trips">https://grouptripper-3c7f1.web.app/trips</a> ) I click on "New trip". A modal opens. I can specify trip title, the modality of collecting dates (either I decide and just ask the rest of participants to accept or decline, or I can ask everyone to first say when they are available, look for common availabilities and finalized dates when everyone responded), enter my date availability/ies, specify how many working days and total days I am available. When I click 'Save', the trip is saved to the database, and the page <a href="https://grouptripper-3c7f1.web.app/trips">https://grouptripper-3c7f1.web.app/trips</a> refreshes, listing the newly created trip. An image is chosen as cover picture for the trip, based on the trip name.	Pass
10	Abandon creation of new plan	I am logged in. From the home page ( <a href="https://grouptripper-3c7f1.web.app/trips">https://grouptripper-3c7f1.web.app/trips</a> ) I click on "New trip". A modal opens. I can close it by clicking on 'Cancel'.	Pass
11	Display existing plans on the home page	<ul style="list-style-type: none"> <li>I am logged in and visit <a href="https://grouptripper-3c7f1.web.app/trips">https://grouptripper-3c7f1.web.app/trips</a>. I can see a card for each trip I have created. If I click on the drop-down menu at the top-left of the page, I can decide to filter upcoming trips, all trips, or past trips. By default only upcoming trips are displayed.</li> </ul>	Pass

12	Delete plans from the home page	<ul style="list-style-type: none"> <li>Each plan card visible here <a href="https://grouptripper-3c7f1.web.app/trips">https://grouptripper-3c7f1.web.app/trips</a> also contains a small delete icon.</li> <li>When I click on the delete icon, a pop-up window asks me to confirm deletion.</li> <li>If I click on 'Yes, delete', the plan and all its related info are removed from the database. The page reloads and the deleted trip is no longer there.</li> </ul>	Pass
13	View details of existing plan	I am logged in. From <a href="https://grouptripper-3c7f1.web.app/trips">https://grouptripper-3c7f1.web.app/trips</a> I can click on any trip card and a dedicated page loads, displaying the trip details.	Pass
14	View/edit participants of plan	I am logged in. I visit <a href="https://grouptripper-3c7f1.web.app/trips/&lt;trip ID&gt;/participants">https://grouptripper-3c7f1.web.app/trips/&lt;trip ID&gt;/participants</a> following steps of test 13. I click on 'Add friends' button in the top-left corner of the participants tab. The required steps to add friends are listed. Step 1) asks to send the sign up link to a friend and ask them to sign up. I will consider this done. Step 2) says: "If you haven't already, add friends to your profile". A button is available to perform this operation. I will assume this step is already done as it is described in more details in test 21. Step 3) invites to type friends names into an input element. I click on the input element and automatically a list of available friends is populated and I can add or remove friends I want to add in one operation before saving. The page reloads and the friend/s I added are displayed with status pending in the list of trip participants.	Pass

15	View / edit itinerary of plan	I am logged in. I visit <a href="https://grouptripper-3c7f1.web.app/trips/&lt;trip ID&gt;/participants">https://grouptripper-3c7f1.web.app/trips/&lt;trip ID&gt;/participants</a> and by clicking on the 'Itinerary' tab I am redirected to <a href="https://grouptripper-3c7f1.web.app/trips/&lt;trip ID&gt;/itinerary">https://grouptripper-3c7f1.web.app/trips/&lt;trip ID&gt;/itinerary</a> . I see a button on the top-left corner of the tab 'Add event'. I click on it, a modal opens, and fill the form to create 6 different types of events. I can save the new event by clicking 'Save'. The page reloads and lists the new event in a card. If I click on the card I can see the event details, modify them and save again. I can also comment on the event. Comments are refreshed immediately on each trip participant's page (websockets channel is open).	Pass
16	View / edit polls for the plan	I am logged in. I visit <a href="https://grouptripper-3c7f1.web.app/trips/&lt;trip ID&gt;/participants">https://grouptripper-3c7f1.web.app/trips/&lt;trip ID&gt;/participants</a> and by clicking on the 'Itinerary' tab I am redirected to <a href="https://grouptripper-3c7f1.web.app/trips/&lt;trip ID&gt;/polls">https://grouptripper-3c7f1.web.app/trips/&lt;trip ID&gt;/polls</a> . I can click on the 'New Poll' button on the top-left corner of the tab and a modal opens, allowing me to create a poll by typing a questions, 2/more answers, deciding how many answers participants can choose. By clicking on the 'Save' button, the page refreshes and I can see the new poll listed on the page. Once the poll is created, all poll participants receive a notification in real time. The notification asks to reply to the poll. By clicking on it, participants can register their answer/s into the database. Each poll card displays a button to see poll results, a button to edit the	Pass

		poll (only available to the poll owner), and a button to comment on the poll. Comments are written and read in real-time from the database via a WebSocket channel opened with the Firebase DB.	
17	delete plan from plan details	I am logged in. I visit <a href="https://grouptripper-3c7f1.web.app/trips/&lt;trip ID&gt;/participants">https://grouptripper-3c7f1.web.app/trips/&lt;trip ID&gt;/participants</a> and by clicking on the 'Settings' tab I am redirected to <a href="https://grouptripper-3c7f1.web.app/trips/&lt;trip ID&gt;/settings">https://grouptripper-3c7f1.web.app/trips/&lt;trip ID&gt;/settings</a> . By clicking on it, plan owners can see a button to delete the trip ('Delete trip'). When clicking on it a pop up alert is displayed. If I confirm, the trip is deleted from the database. All related information is deleted (notifications, clean up trip participants' records).	Pass
18	Remove myself from plan	I am logged in. I visit <a href="https://grouptripper-3c7f1.web.app/trips/&lt;trip ID&gt;/participants">https://grouptripper-3c7f1.web.app/trips/&lt;trip ID&gt;/participants</a> . I can click on button 'Abandon trip' available in the first card on the page. If there are other participants in the trip and I am the owner, I am prompted to remove them first. Otherwise, I can confirm and remove myself from the trip. If I am the owner, the trips is deleted, otherwise not. The DB is cleaned up by removing all information related to this trip if the trip is removed.	Pass
19	Edit trip name	I am logged in. I visit <a href="https://grouptripper-3c7f1.web.app/trips/&lt;trip ID&gt;/participants">https://grouptripper-3c7f1.web.app/trips/&lt;trip ID&gt;/participants</a> . At the top of the page, on the hero image, I can click on the pencil icon and modify the name of the trip. Then I can save it by clicking on the save icon. This operation is available from all trip tabs.	Pass

20	View / edit personal info	<p>I am logged in. I click on the navbar burger menu, 'Profile'. I am redirected to <a href="https://grouptripper-3c7f1.web.app/profile/&lt;user id&gt;">https://grouptripper-3c7f1.web.app/profile/&lt;user id&gt;</a>. From here, I can both change my profile picture and change my username.</p> <p>When I click on 'Edit picture', a modal opens, in which through a button I can select an image from local storage, resize it and center it. I can save the new image by clicking on 'Save' or cancel the operation.</p> <p>To change my username I can click on the pencil icon next to it. This changes the username to editable input box and replaces the pencil icon with a save button. An alert message confirms the success or failure of the operation.</p>	Pass
21	View / edit friends	<p>I am logged in. I click on the navbar burger menu, 'Profile'. I am redirected to <a href="https://grouptripper-3c7f1.web.app/profile/&lt;user id&gt;">https://grouptripper-3c7f1.web.app/profile/&lt;user id&gt;</a>. From here, I can click on the 'Friends' tab. From the search bar I can look for an exact match for a username or email address registered in the database. If no results are found, a corresponding message is displayed. If a result is found, it is returned with a corresponding available action (none if user is myself, none if user is already friend, 'add' if I can add to friends). When I click the add button, a notification is immediately sent to the target user. They can accept or refuse my friendship request and as soon as they do, I am in turn notified. Once two users are friends, any of them can remove the other from their friends. The</p>	Pass

		other user is notified. On this tab I can see my list of friends, divided by accepted and pending.	
22	Change language	<p>The system must allow users to specify the language they prefer in a visible position on the navbar.</p> <p>The system must translate all content upon selection of a language</p>	Not implemented – outside scope of MVP's acceptance criteria
23	Map view	The system must display a map with pins for each location listed as a desired itinerary	Not implemented – outside scope of MVP's acceptance criteria
24	Allow to make plans public	<ul style="list-style-type: none"> <li>• The system must allow users to make their plans public</li> <li>• The system must allow users to browse existing public plans filtering by location and days</li> <li>• The system must allow to copy an existing template to new and edit it</li> </ul>	Not implemented – outside scope of MVP's acceptance criteria

## Appendix T – Security scan results

### T.1 results view – first scan

The screenshot shows the HostedScan.com web application interface. At the top, there is a navigation bar with links for Dashboard, Targets, Scans, Risks, Reports, Pricing, Docs, and a user profile icon. Below the navigation bar is a header with a shield logo, the text "HostedScan", and buttons for "Add Targets" and "New Scan".

**Scans**

A table displays three completed scans:

SCAN TYPE	TARGET(S)	STATE	PROGRESS / RESULTS	CREATED	ACTIONS
OpenVAS	<a href="https://grouptripper-3c7f1.web.app/">https://grouptripper-3c7f1.web.app/</a>	Succeeded	<a href="#">PDF</a> <a href="#">XML</a>	14 minutes ago	
OWASP ZAP	<a href="https://grouptripper-3c7f1.web.app/">https://grouptripper-3c7f1.web.app/</a>	Succeeded	<a href="#">PDF</a> <a href="#">HTML</a> <a href="#">JSON</a> <a href="#">XML</a>	14 minutes ago	
Nmap	<a href="https://grouptripper-3c7f1.web.app/">https://grouptripper-3c7f1.web.app/</a>	Succeeded	<a href="#">PDF</a> <a href="#">HTML</a> <a href="#">JSON</a> <a href="#">XML</a>	14 minutes ago	

Below the table, there are pagination controls: "Rows per page: 10 ▾", "1-3 of 3", and navigation arrows.

**Scheduled Scans**

A table displays three scheduled scans:

TARGET(S)	SCAN TYPE	SCHEDULE	LAST RAN	NEXT RUN	ACTIONS
(all targets)	Nmap	Monthly		in 29 days	
(all targets)	OpenVAS	Monthly		in 29 days	
(all targets)	OWASP ZAP	Monthly		in 29 days	

## T.2 OpenVAS—first scan

Scan report - ! 2 new risks detected by OpenVAS on https://grouptripper-3c7f1.web.app/  

HostedScan.com <no-reply@hostedscan.com>  
to me ▾

5:06 PM (1 hour ago)



This email contains a summary of your OpenVAS report. The complete OpenVAS report is also attached as a PDF.

### Risks Detected

Total number of unaccepted risk occurrences by severity.



### New Risks (2)

New risks detected since your last scan.

ICMP Timestamp Reply Information Disclosure (1 occurrence)	 Low
TCP Timestamps Information Disclosure (1 occurrence)	 Low

### T.3 Nmap – first scan

Scan report - ! 2 new risks detected by Nmap on https://grouptripper-3c7f1.web.app/  

HostedScan.com <no-reply@hostedscan.com>  
to me 

4:53PM (1 hour ago)



This email contains a summary of your Nmap report. The complete Nmap report is also attached as a PDF.

#### Risks Detected

Total number of unaccepted risk occurrences by severity.



#### New Risks (2)

New risks detected since your last scan.

---

Open TCP Port: 443 (1 occurrence)  Medium

---

Open TCP Port: 80 (1 occurrence)  Medium

#### T.4 OWASP— first scan

Scan report - ! 10 new risks detected by OWASP ZAP on https://grouptripper-3c7f1.web.app/ ➝ [Inbox x](#)

HostedScan.com <no-reply@hostedsan.com>  
to me ▾

4:53 PM (1 hour ago) ★ ↵

## OWASP ZAP Report

by HostedScan Security



This email contains a summary of your OWASP ZAP report. The complete OWASP ZAP report is also attached as a PDF.

### Risks Detected

Total number of unaccepted risk occurrences by severity.

0 Critical	0 High	3 Medium	7 Low
------------	--------	----------	-------

### New Risks (5 of 10)

New risks detected since your last scan.

---

Absence of Anti-CSRF Tokens (1 occurrence)	<span style="color: orange;">■</span> Medium
Content Security Policy (CSP) Header Not Set (1 occurrence)	<span style="color: orange;">■</span> Medium
Missing Anti-clickjacking Header (1 occurrence)	<span style="color: orange;">■</span> Medium
Cookie Without Secure Flag (1 occurrence)	<span style="color: blue;">■</span> Low
Cross-Domain JavaScript Source File Inclusion (1 occurrence)	<span style="color: blue;">■</span> Low

## T.5 – second scan

**1**

Scans in progress

**3**

Scheduled scans

Risks Breakdown

Risk Level	Count
Critical	1
High	10
Medium	13
Low	14
Accepted	1
Closed	2

**Most Recent Scans**

SCAN TYPE	TARGET(S)	PROGRESS / RESULTS
Sslyze	https://grouptripper-...	<a href="#">PDF</a>
OWASP ZAP Active	https://grouptripper-...	<a href="#">PDF</a>
Nmap UDP	https://grouptripper-...	<a href="#">PDF</a>
OpenVAS	https://grouptripper-...	<a href="#">72%</a>
OWASP ZAP	https://grouptripper-...	<a href="#">PDF</a>

[See all 12 scans](#)

**Most Recent Risks**

TITLE	TARGET	THREAT LEVEL / STATUS
Absence of Anti-CSRF Tokens	https://grouptripper-3c7f...	Medium
CSP: Wildcard Directive	https://grouptripper-3c7f...	Medium
CSP: style-src unsafe-inline	https://grouptripper-3c7f...	Medium
Content Security Policy (CSP) Header Not Set	https://grouptripper-3c7f...	Medium
Cookie No HttpOnly Flag	https://grouptripper-3c7f...	Low

[See all 26 risks](#)

>	Missing Anti-clickjacking Header	OWASP ZAP	https://grouptripper-3c7f1.web.app/	<span style="color: orange;">■</span> Medium	<span style="background-color: #ccc; border-radius: 50%; padding: 2px 5px;"></span> Closed	49 minutes ago	2 hours ago
>	Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	OWASP ZAP	https://grouptripper-3c7f1.web.app/	<span style="color: blue;">■</span> Low	<span style="background-color: #ccc; border-radius: 50%; padding: 2px 5px;"></span> Closed	49 minutes ago	2 hours ago