



UTrip

Organize your trip today!

Umeå University

Databasteknik (5DV187)

Simon Triscos, Jose Manuel Rus Manjón, Salome Müller

October 28, 2021

### **Abstract**

**UTrip** is a web-application that allows you to plan trips of any kind in an easy fashion. Sign up using your email, invite friends and create trips you always wanted to go on. We will provide you with forms where you and your friends can, simultaneously, add the resources you need and assign them to each other. Once you start packing for the trip, you see all resources you should bring in your own checklist. Here you can also add personal things. Checkoff everything you packed, and make sure you don't forget anything. Enjoy your trip!

# 1 Introduction

**UTrip** is a web-application, that lets you organize your trip in an easy manner. You can create a trip, specifying the location, kind, start date and duration. We provide you with some common kinds of trips. If you choose one of them, your trip's resources page will automatically contain some resources that you should not forget. Of course you can add anything you want later on. Once the trip is created, you are able to invite users via their email.

The trip has multiple forms, a general one and four additional ones for each day : one for breakfast, lunch and dinner, as well as one for the equipment you need to spend the night. This way you will have a good overview of what you need, and what nobody has taken care of yet. If you plan on staying somewhere, simply add the name of the refuge and assign someone who should book it.

All forms that have resources which are not yet assigned are colored in red. This way you are able to see with one glance where your attention is needed.

When you start packing for the trip, you can use the checklist to make sure that you will forget nothing. There, you see all the items you should bring, and can add as many private items as you want. Once you packed something, simply check it off the list.

# 2 Instructions

The following subsections are a step-by-step guide on how to setup **UTrip** and how to use it.

## 2.1 Installation

If you want to download and run our web-application, download or clone the code from [GitHub](#). The repository contains a ReadMe with the instructions on how to get the application to run on a Windows based operating system.

## 2.2 Sign-up and Login

First of all, you need to sign in. You can do this by clicking on **signin**. You will be redirected to a page, where you can enter all your information. You should provide a username which is longer than 8 characters, a valid email address, and a strong password (use uppercase and lowercase letters,

numbers, symbols and more than 8 characters). After signing in you are

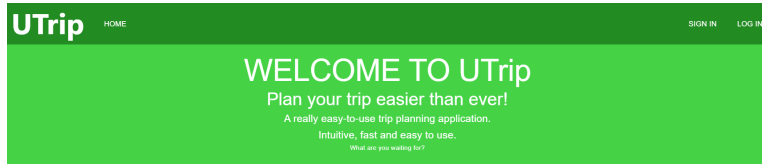


Figure 1: Homepage before logging in

redirected to the **account** page. When you are logged in, the navigation bar contains **trips**, **checklist** and **account**.

## 2.3 Create a trip

On the **trips** page you can see all your upcoming trips. Here, you can either choose a trip to edit (see next section) or create a new trip.

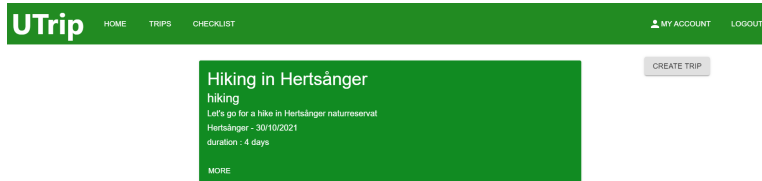


Figure 2: All upcoming trips

When creating a trip, you see the form shown in Figure 3. You are asked to give your trip a title, description, location and start date, duration and kind. Based on the defined duration and kind, the content you see when editing the trip differs. For each kind (despite *other*) the trip's general form

Figure 3: Form for creating a trip

will have specific predefined fields. For *hiking* these are things like a sleeping bag and a tent, whereas for *climbing* you should take your helmet and your climbing shoes. The duration defines for how many days you will have forms to add food and equipment.

## 2.4 Edit a trip

When you are on the **trips** page and click on **more** you will reach the specific trip's page. Here, you can fill the forms or edit the trip. The trip can also be finished once it is over, so it will no longer be listed on your **trips** page. With the **invite** button you can invite friends via their email. Be sure to type the email they used to sign up to **UTrip**.

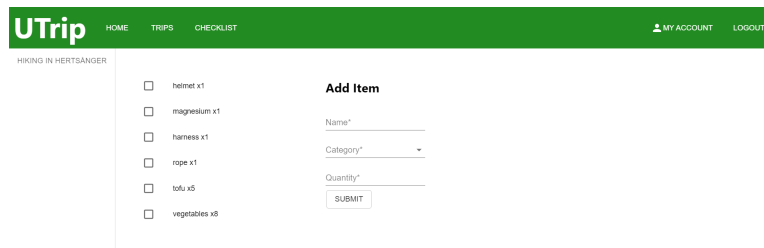
Figure 4: Form to edit a trip

The main part of the page is devoted to the trips form. On the left you

see the general form and the days of the trip. When a day is selected, you can see the separate categories on the right. Notice that forms that have resources which were not yet assigned are colored in red. For each resource you can choose one of the participating users to bring this item. On the bottom you find the button to add more fields.

## 2.5 Checklist

On the checklist page you can see all your trips listed. For each of the trips you can see a to-do-list with all the resources that you are responsible for. Here you can also add personal items that you want to bring with you when starting the trip.



The screenshot shows the UTrip web application interface. At the top is a green navigation bar with the 'UTrip' logo and links for 'HOME', 'TRIPS', and 'CHECKLIST'. On the right of the bar are links for 'MY ACCOUNT' and 'LOGOUT'. Below the navigation bar, the page title is 'HIKING IN HERTSANGER'. The main content area is divided into two columns. The left column contains a list of resources with checkboxes: 'helmet x1', 'magnesium x1', 'harness x1', 'rope x1', 'tofu x5', and 'vegetables x5'. The right column is titled 'Add Item' and contains a form with three input fields: 'Name\*', 'Category\*' (a dropdown menu), and 'Quantity\*'. A 'SUBMIT' button is located at the bottom of the form.

Figure 5: Checklist for the trip

## 3 Implementation

Our web-application is based on React with Flask as server and a PostgreSQL database.

### 3.1 Backend

The Flask framework enhanced by some libraries, i.e., psycopg2, CORS and praetorian.

psycopg2 is used to access the database. After connecting to the database, it provides functions to execute SQL statements and fetch the results. It provides thread safety, assuring that concurrent update, insert and delete operations do not interfere with each other. It further supports establishing

and closing a lot of connections to the database in a short period of time, and is thus ideal to use with Flask, where a new connection must be established for every access.

CORS is a python library, which allows Cross Origin Resource Sharing. Without CORS, requests from clients would not be answered, in order to prevent attacks from malicious scripts.

User authentication is done with `flask-praetorian`. The `praetorian` package contains functions to create JWT tokens of a users data. When logging in, the client gets this token from the user. For protected routes, the token must be sent in the header of the HTTP request. `Praetorian` relies on some basic `sqlAlchemy` functionality, which is why this library is also used.

## 3.2 Frontend

The frontend was implemented with JavaScript and React JS, using Material UI to improve the appearance of the web app, and some other libraries like `react-router` to manage pages routing.

React is known for its speed and useful characteristics like the virtual DOM and the possibility to inherit from parents components making it a safe, fast and easy to use front-end framework.

In addition, we are using token authorization by `react-token-auth` together with the previously mentioned `flask-praetorian` to manage sessions and avoid any kind of unauthorized access, making our web app safe and preventing impersonation attempts.

The basic structure of the React app is a main component containing the navigation bar, and a switch to different routes. Each route leads to another principle component, which represent the content of each page. When necessary, for example for pages detailing a specific trip, the trip's ID is passed in the route URL. Clicking on a navigation button changes the current URL, thus loading the component attached to the corresponding route.

Our project uses a fair amount of forms, some of which, for example the resources forms in the trip-specific pages, need their format changed at run time. To easily define and manipulate forms, we used `JSON Forms`, a solution based on the use of schemata, and well supported on React.

Finally, to communicate with the backend, we used `Axios`, a promise-based HTTP client for NodeJS. `Axios` allowed us to send HTTP request to the backend, receive data from it, and handle errors.