

Ideas of further development

This document's purpose is to describe and partly estimate ideas of further development. These ideas came from the customers of the project, Juuso and Arto or the end users of the app during the project implementation. Due to limited amount of time the ideas were not possible to be implemented in autumn 2021. All these ideas would improve the usability of the application or bring more value to the users.

Live sharing the location of a user

Description of the idea

Live sharing of the location of a user would be very valuable safety wise. Location sharing would be like the features offered by, for example, Whatsapp and Google Maps. The user could start the live share from the application when they start their trip in Pallas area. The live location would be either accessible by a friend that the user sends the location link to, or by a person/people in the area. The location could then be used by rescue people if an accident happens.

Estimation of implementation

This idea was first accepted as an item for the project as it would bring a lot of value to the user. However, the team would have not had time to implement anything more complicated than showing the current coordinates of the user in a way that the user could copy them and forward them to someone. As this implementation would not have helped in emergencies at all, we decided not to implement it.

Implementing the live location sharing within the application was quickly noticed to be too complicated. Some of the questions and issues regarding the implementation are:

- How could we access the location of the user even if the web application is not actively open?
- What permissions would be needed for accessing, storing and sharing the location?
- Would the data be considered personal? How would GDPR be considered?
- How would we store the location?
- How well or accurately can the location be accessed in the remote Pallas region?

We considered other simpler possibilities of implementing the functionality, too. We tried to come up with a way to utilise the live location sharing of another application through a button in Snowledge. Whatsapp does not offer live location through its API, neither does Google Maps. The only possibility we come up with is implementing a Telegram bot that requests the user to send their live location to it (<https://core.telegram.org/bots/api#location>). The link to the bot would be somewhere in the application. We did not research how the data could be then handled and thought that Telegram is not a good application to use since the users of Snowledge do not all use it.

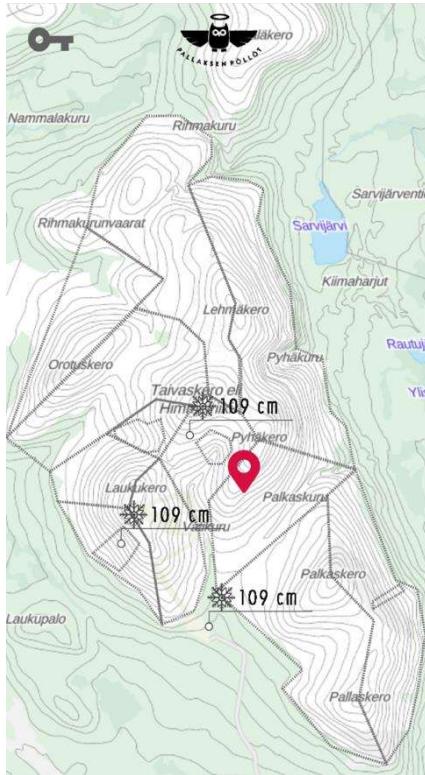
After examining using other applications we concluded that it would require a lot of effort to come up with a solution and the possible solutions would not bring more value to the customer. We decided to include a reminder for the user to share their location in the welcome text of the application in this version of the application.

Snow depth measurement points

Description of the idea

Juuso described the need to give the users more information regarding the depth of the snow in the area. This information could be given for certain points on the map as seen in picture 1.

The information of each snow measurement point could be either manually entered by the admins or through with the new sr50a-equipment. We did not discuss what the admin view of adding snow depth data could look like. The user could see the most recent snow depth record by viewing the measurement points on the map.



Picture 1: Snow depth measurement points on the application's map

Estimation of implementation

Implementing this functionality would include implementing the data storage of the snow measurements, a way to input new data and a way to show the data to the user. This would be a brand-new functionality to the application.

The measurement points should be created into the application, added to the map and have corresponding information views. The snow depth information could be fetched from the database and shown in the same way as the segment information. The points could be added in the same way as segments (their names and coordinates would be defined in the database and they would be drawn on the map) or as icons floating on top of the map.

The existing SQL-database could be used to store the snow measurement data. A new table could be created for the snow depth measurements, each measurement station would then query for the most recent data for them. The Pöllöt could also use the data to see the snow measurement info of a certain time scale.

The snow measurement table would be very simple with no necessary dependencies on the other tables. The way the data is linked to the measurement points depends on how the points are implemented. It is not needed to worry about filling the database with a lot of snow depth information since the information is so simple. The table could look like this:

LUMEN_SYVYYDET:

```
INT mittauspiste_ID
```

```
INT syvyys
```

```
DATE mittausaika
```

One way to implement the admin view, would be following the same principle as with the segment information. The admin could see a hidden “Edit” button in the snow depth information view when they are logged in and access a floating window for entering new data. This window would only require one input from the user: the depth and after saving that information would be inserted to the SQL-database. This would require less work than it did to create the snow record entry view as it is simpler and an example already exists in the codebase.

The snow depth data could also be inserted by the SR50A equipment automatically. Most likely that can be integrated to enter data to the SQL-database but we did not research this possibility.

To summarise, there are many ways of implementing the measurement points. Most likely that can be implemented in a way that is similar to the already existing components for showing and entering segment information. The data is simple and inserting and fetching in from the SQL-database could be less than a day’s work especially since there are existing examples with the segment information. In total developing this totally new functionality would take over a week.

Using the new SR50A snow measurement equipment

Description of the idea

A SR50A Sonic Ranging Sensor snow depth measurement equipment will be installed in Pallas region in winter 2021. Ilmatieteenlaitos uses this same equipment. The data from the equipment could be used for one of the snow depth measurement points. The team did not research the usage or integration of the equipment. We were unable to find easily readable documentation or instructions of the equipment.

Using data from Ecowitt.net

Description of the idea

The current weather service in Pallas sends data to a website called Ecowitt.net. The weather station has been collecting rain information until May 2021 and the station will be taken back in to use soon. Using this information for the application would ensure that the application’s weather data is accurate.

Estimation of implementation

Ecowitt does not provide a REST API where the weather data could be requested from. According to this forum, it seems like an API will be available soon:

<https://www.wxforum.net/index.php?topic=42108.0> The API could then be used to query for the

weather station data in JSON format. No detailed descriptions of the API have been released and the schedule is still unclear.

Once the API is available, using the weather station's data can be straight forward. The application needs to query wanted data from the API that can then be used in the weather tab information or in the new snow depth measurement points. There will be no data storage requirements since the information can be queried whenever its needed.

This gateway was also looked into: https://github.com/iz0qwm/ecowitt_http_gateway. It seems like using this would require a significant amount of work, especially regarding building the environment.

Collecting donations from the users

Description of the idea

The application could have an information box that asks the user for a donation to support the development of the application. The information box could be, for example, a popup with the text "*Hei, haluatko tukea Pöllöjen työtä lumisovelluksen eteen ja varmistaa sen jatkokehittämistä? Lahjoita 10€ tästä.*". The popup should appear only once per user.

The tool or channel for the donation has not been discussed. Some possibilities are listed below. One or both could be used.

- MobilePay:
 - o The user would be given a 5-digit donation number that the user can then use in MobilePay application to donate
 - o The users would need to have a MobilePay account and application
- Bank account information:
 - o The user would be given a bank account number and possibly other transaction details
 - o The users would not need other applications, but this solution would require more effort from the user

It should be noted that collecting money from the users of the application requires a permission. There are two different types of permissions that can apply in this case: *Rahankeräys* and *Pienkeräys* (maximum 10 000€ and 3 months duration). More information is available here: <https://polisi.fi/rahankeraykset>

Estimation of the implementation

The implementation of the donation would not require much development work. Most work would be done setting up the donation channel and permissions. The only needed implementation to the application would be a popup component or other way of showing the user the donation information. This would require less than a day's amount of work.

If MobilePay is chosen as a channel for the donations a MyShop should be created for the collection. The starting fee is 99€ and with less than 2000 events MobilePay takes 0,75% of the transactions.

The pricing information is available here

<https://www.mobilepay.fi/yrityksille/myyntipisteet/mobilepay-myshop#myshophinnat>.

Admins can enter skiables when entering new segment info

Description of the idea

Since the skiability of a snow depends on other factors than just the snow type, it would be valuable if the admins could set the skiables when editing segments. This could be implemented by adding skiability as one of the inputs required from the admin in the snow record entry view. The skiability could be added for each snow type and the default value of the skiability would be the scores set in the database.

Estimation of the implementation

The implementation would be straight forward. The new inputs would be added to the snow record view and the view would have to fetch the default skiability scores from the database. The database would be edited in a way that the snow update (Paivitys) would have columns skiability1, skiability2 etc. This would require minor changes in the creation of the Paivitys-table, router functions and the insert-statement of storing the new segment information.

The snow record would have to be edited minorly, too. The skiability of each of the snow types would not be fetched from the Lumityyppi-table but from the Paivitykset-table. This might be simpler than the current implementation is. We estimated that this functionality would take about 15 hours to implement.

Editing the warnings in the information tab

Description of the idea

Now the application has one non-changeable warning on the information tab. In reality the conditions in the area are always changing and it would be useful if the admins could add or edit the warnings in the information tab.



Picture 2: Snow type description tab with warnings

Estimation of the implementation

Currently the snow type description tab has one avalanche warning and the descriptions of all snow types. These are hard-coded in the file `src/map-app/src/SnowTypes.js`. With this implementation, it is possible to add new information to the tab by editing `SnowTypes.js` file.

A warning looks like this in the file:

```
<Box className={styledClasses.avalanche}>  
  </img>  
  <Typography variant="h4">Lumivyöryvaroitus</Typography>  
  <Typography>  
    Pallaksen lumivyörymaastoja ovat useat kurut kuten Pyhäkuru,  
    Palkaskuru...  
  </Typography>  
</Box>
```

Copying this text and pasting it right underneath the original text would create a new warning. The name of the warning and its description are highlighted in blue, those are the only parts that need to

be changed when adding a new warning. The icon is set with the line. This can be erased or a new icon can be added to the icons folder and its name can be added as the “img src”. The instructions written by the team for the customer has more detailed instructions on how to make minor code changes and deploy them on the server.

If there are changes to the warnings only occasionally this might be a sufficient workaround to adding new warnings. If the warnings should be edited more often a better solution should be developed. The work would include editing the snow description tab so that it queries the information from the database and designing implementing an admin view for adding or editing the warnings that would then be connected to the database. Our estimated effort for this work is about a week.

Feedback boxes

Description of the idea

There are two kinds of feedback boxes that we discussed. One would be a feedback box for the users that would store their feedback somewhere where only the admins or Pöllöt could access it.

Another discussed idea was making the application more interactive and allowing the users to send their information of segments' snow situation. This would either be visible as kind of a chat view in the segments for all users or only as messages to the admins. The admins could then update the snow information if needed.

This idea was only briefly discussed and we did not design it any further. Getting feedback from the users is a high priority if the customer continues working on the application. Gathering feedback could be as simple as providing an email address somewhere in the application or asking for it in the hotel where the application is in use.

Search for snow type legends in the information tab

Description of the idea

Now the list of snow types in the information tab is quite long. The user's only way to look for a certain snow type is by going through the whole list. The search of a certain type would be faster if the tab had a search functionality or if the snow types were divided into categories that could be extended. This would improve the usability of the tab, however based on the team's and customers' discussion searching for a certain snow type is not a very common or important use case.

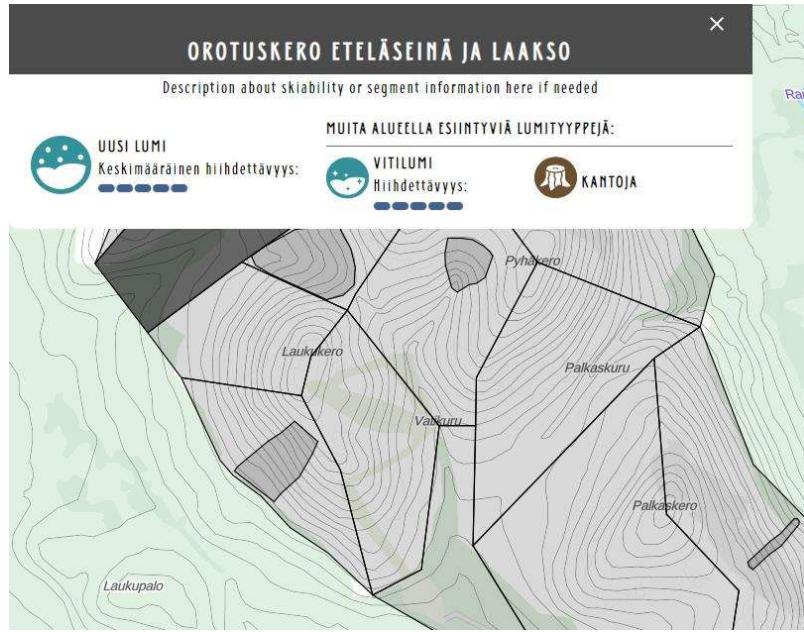
Estimation of the implementation

This functionality would be simple to implement. The snow types have category_IDs in the database and can then be easily divided into categories like they are in the drop down in the snow record entry view. The entry view also has an example of a search bar that could be used. Adding one of these options would take a few hours to implement.

Moving the map so the map is never hidden by snow record view

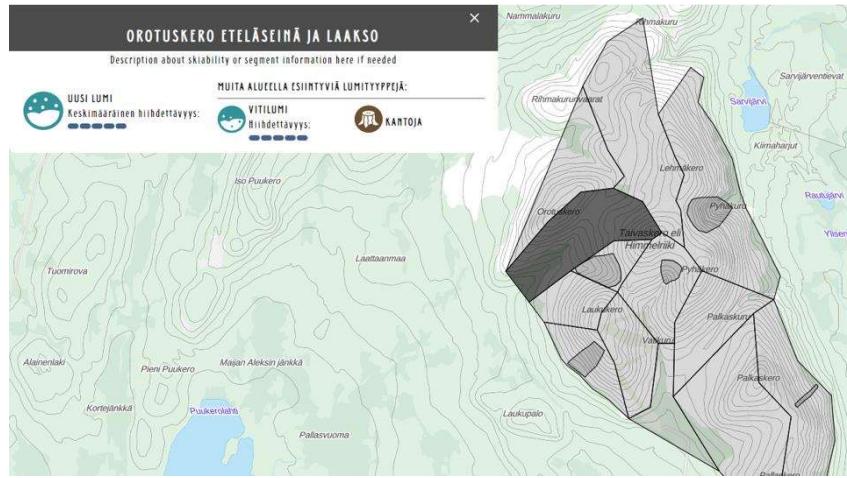
Description of the idea

At this state the map can be even largely hidden by the segment information. An example can be seen in the image below. This issue was not classified as major based on the usability testing.



Picture 3: Map hidden by segment information

The map could be easier to use and access if the map would be moved in a way that it is never hidden under the segment information. Especially on the web-view there would be plenty of space for the fell area to be moved to when a segment has been opened, like in the picture below.



Picture 4: Map moved away from the segment information

Moving the important region of the map away from the segment information has pros and cons. No important parts of the map would be hidden by the segment, but the user might get confused with the map suddenly moving. In the Pallas region the map is takes more space vertically than horizontally, making it easy to move. However, if the application is taken into use in other regions, the shape of the important regions might be very different.

Another way of solving the issue would be placing or showing the segment information in a different way. For example, the segment information could be shown in a separate area instead of on top of the map.

[Showing the location of the hotel as an icon on the map](#)

A user taking part in the usability testing mentioned that the map would be more easily understandable if the location of the hotel was more clearly indicated on the map. The location could be shown with an icon placed on top of the map at the hotel's location. An icon for the hotel should be designed. Placing it on the map would probably take just a few hours.

[Changing the font used in the application](#)

The applications font Donau was chosen as it has been used by Pallaksen Pöllöt in other materials, too. The font was not tested by users. The usability experts of the team voiced their concerns that the font might be difficult to read by some users. In the future, the readability of the font could be evaluated, and the font possibly changed. The font can be easily changed by changing the font-family from styling files.