FINAL REPORT

Mobile App for Diabetes Type 1 Monitoring Developed for Improved UX

Group #15

Aigerim Tushieva, Yussufkadir Syurmen

Adviser: Khalil Khan

April, 2024

Executive Summary

Our project aimed at improving functionalities of mobile apps dedicated to people with diabetes to help users with their diabetes management and decrease attrition rate. Ultimately, we developed a cross-platform mobile app with three completely novel features: drag & drop data logging, incorporated an ML model for visualization of diabetes data and a rule-based chatbot for monitoring the diabetes progress. Finally, we evaluated the app by collecting responses from a person with T1D (one of the team members' younger brother, who is 19 years old) on a questionnaire.

Introduction

Diabetes Type 1 (T1D) is a chronic disease, in which the pancreas is partially or completely incapable of producing insulin — a hormone that controls the glucose balance in the organism. Currently, T1D doesn't have a cure, but a treatment and well-maintained diabetes management can ensure a high quality of life. There are over 100 apps for T1D in Google Play and App Store, which are aimed at improving users diabetes management by allowing users to log their blood glucose (BG) readings and monitor statistics. However there are three problems associated with them:

1. None of the popular apps incorporate machine learning models or/and chatbots for UI/UX to improve diabetes management.
2. None of the popular apps target a condition called Nocturnal Hypoglycemia (NH) — an episode of low BG level during night sleep, which renders this condition particularly dangerous (Islam, 2021).
3. Diabetes data logging functionality — the most used feature in these apps — is not optimized and slow, which causes high attrition rates (Arnhold and Quade, 2019).

Our solutions to address the above problems were set as follows:

1. Develop a cross-platform mobile app
2. Train and incorporate a ML model for visualization of BG and implement a rule-based chatbot for improved monitoring of statistics.
3. Train a model to classify a possible NH given a set of features of prior BG values and other diabetes related data and integrate it to the mobile app
4. Create a UI for logging BG readings where a user doesn't need to type in values and time, but instead drag & drop BG values on a flexible timeline with pre-set time intervals, which can be expanded or shrunk using finger swipes.

Organization the report

1. Background work on NH classification
2. Project approach/execution/evaluation consists of two parts covering NH task and mobile app tasks.
   a. Both parts cover approaches, how they were executed and evaluated, the issues encountered and the decisions taken to overcome them.
   b. The part covering the mobile app has 3 subparts on ML visualization task, rule-based chatbot task, and data logging task.
3. Links to all of the source codes are indicated after the Conclusion.
4. References

5. Appendices with additional screenshots of the mobile app and questionnaire

Background Work

Prediction of NH is one of the topics of ML research in diabetes, though it isn't represented by as many studies as general prediction of BG. For example, Bertachi et al (2020) used MLP and SVM, where the latter showed higher accuracy with sensitivity and specificity of 78.75% and 82.15%. The authors collected data on CGM, carb intake, insulin injections, physical activity from 8 patients over 12 weeks, and the main selected features consisted of bedtime BG, variability of BG prior to sleep, bedtime slope of BG over 30 minute, insulin on board (accumulated active insulin in body), carbs on board, accumulated physical activity, and mean BGs 1, 2, 3, 4 hours prior to sleep. Target class (episode of NH) was Class 1 if in the 6 hour period after onset of sleep time BG dropped below 3.9 mmol/L. Parcerisas et al (2022) trained a SVM model to classify NH or no-NH events and achieved 0.70 of sensitivity and a specificity of 0.73. The authors followed the methodology of Bertachi et al (2020), collecting the same data features from 10 patients over 12 weeks, and setting the same requirements target class. The study by Calhoun et al (2020) factors associated with NH were determined using RMRF: daytime hypoglycemia, bedtime BG, HbA1c, IOB (insulin on board, which is the measure of active insulin in body), and physical activity, all listed with decrease of their variable importance. Hence, the studies by Bertachi et al (2020), Parcerisas et al (2022), Calhoun et al (2020) and also by Jensen et al (2020) (forward selection resulted in sensitivity and a specificity of 0.75 and 0.70) all proved the feasibility of prediction of NH with an accuracy around 0.70.

Project Approach, Execution, Evaluation

Nocturnal Hypoglycemia

We had a binary classification problem: given a set of features N we needed to predict whether it is of Class 1 (night with NH) or Class 0 (night with no NH). The reference studies Bertachi et al (2020), Parcerisas et al (2022), Calhoun et al (2020) and Jensen et al (2020) used the following features:

1. Bedtime BG, and mean BGs 1, 2, 3, and 4 hour prior to time sleep time
2. Slope of BG over 30 minute time interval prior to sleep time
3. IOB (insulin on board), accumulated active insulin in body
4. COB (carbohydrates on board), accumulated active carbs in body
5. Physical activity (readings from fitness bands and number of steps)
6. Patient-level data (BMI, HbA1c)

The only dataset that covered above feature requirements and was publicly available was OhioT1DM Dataset[1], which required signing a Data Use Agreement by the research institution. However, it was rejected for our use, so instead we draw our attention to the datasets generated from proof-evaluations of CGM device performance. The dataset was from a study[2] "A Randomized Trial Comparing Continuous Glucose Monitoring With and Without Routine Blood Glucose Monitoring in Adults with Type 1 Diabetes". It satisfied the requirements, as included only patients with T1D, had in total 100+ participants and besides 14M rows of CGM data it included pump bolus data (insulin injections with timestamps) and information on height, weight, and HbA1c.

[1] Link to the OhioT1DM Dataset http://smarthealth.cs.ohio.edu/OhioT1DM-dataset.html
[2] Link to the dataset used https://public.jaeb.org/datasets/diabetes

Our initial features were:

1. Mean BG readings at 23:00, 22:00, 21:00, 20:00
2. IOB (sum of insulin doses taken throughout the day)
3. HbA1c
4. BMI

For all the parts of the ML workflow we used Colab and scikit-learn library. The most time-consuming part of the ML part was data cleaning and feature engineering, because the initial dataset was a collection of BG measurements or insulin doses with timestamps and patient's ID and contained many time gaps, since patients didn't use them regularly and frequently skipped days and took them off for several hours. Furthermore, patients preferred to put on the CGM device during nighttime (which was motivated by their fear of NH), so it rendered many rows of data unusable. Due to that, after completing data cleaning and matching overnight BG readings (containing target value) with preceding daytime BG readings (containing feature values) 35K rows instances of profiles.

We trained RF, SVC models from scikit-learn and keras's MLP, and all models showed sensitivity in the range of 0.1-0.2 with specificity around 0.95 independent of any tuning of hyperparameters.

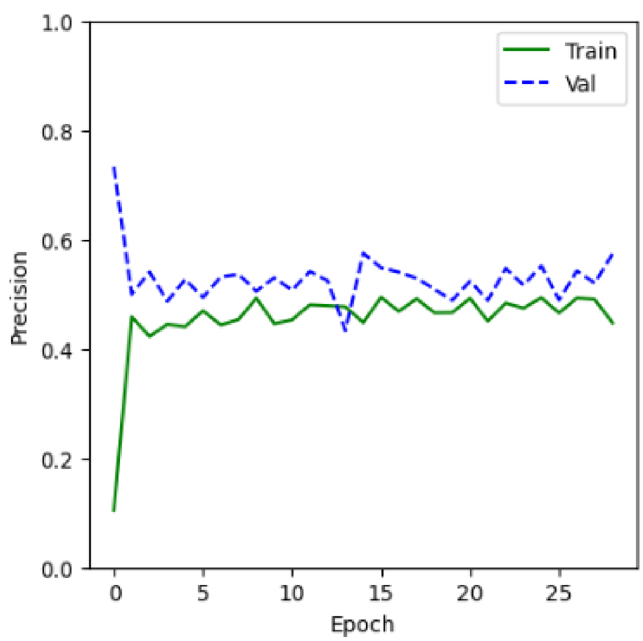


Figure 1. Performance of the MLP classifier on the second dataset

At first, such a poor performance drew suspicions to the dataset, and we found out that the authors of the dataset included a time verification document, where patients made visits to the researchers to verify that their device time was consistent with local time. However, the visits were made once every 3-5 weeks (the study lasted 26 weeks), so it wasn't clear how we could correct the timestamps. Instead we decided to remove all profiles corresponding to days where patients make insulin injections after 23:00 and during the night (before 06:00 in the morning) — it guaranteed us the profile had correct timing, as it's more likely for people with T1D to do insulin injection in daytime. So 35K rows yielded 15K rows (we also excluded CGM readings if there were no insulin data for the corresponding day). However, while training models in this new dataset we still had a very low sensitivity and high specificity of 0.2 and 0.9, which drew our attention to the representation of target values in the dataset, and it turned out that there were only 6.5% of NH episodes, which made our dataset imbalanced and in fact was inconsistent with Jensen et al (2020), where NH occurred every fifth night. We examined another dataset ("A Randomized Trial to Assess the Effectiveness of the GlucoWatch Biographer in the Management of Type 1 Diabetes in Children" with 200K rows of CGM data[3]) and there NH occurred in 22% of nights.

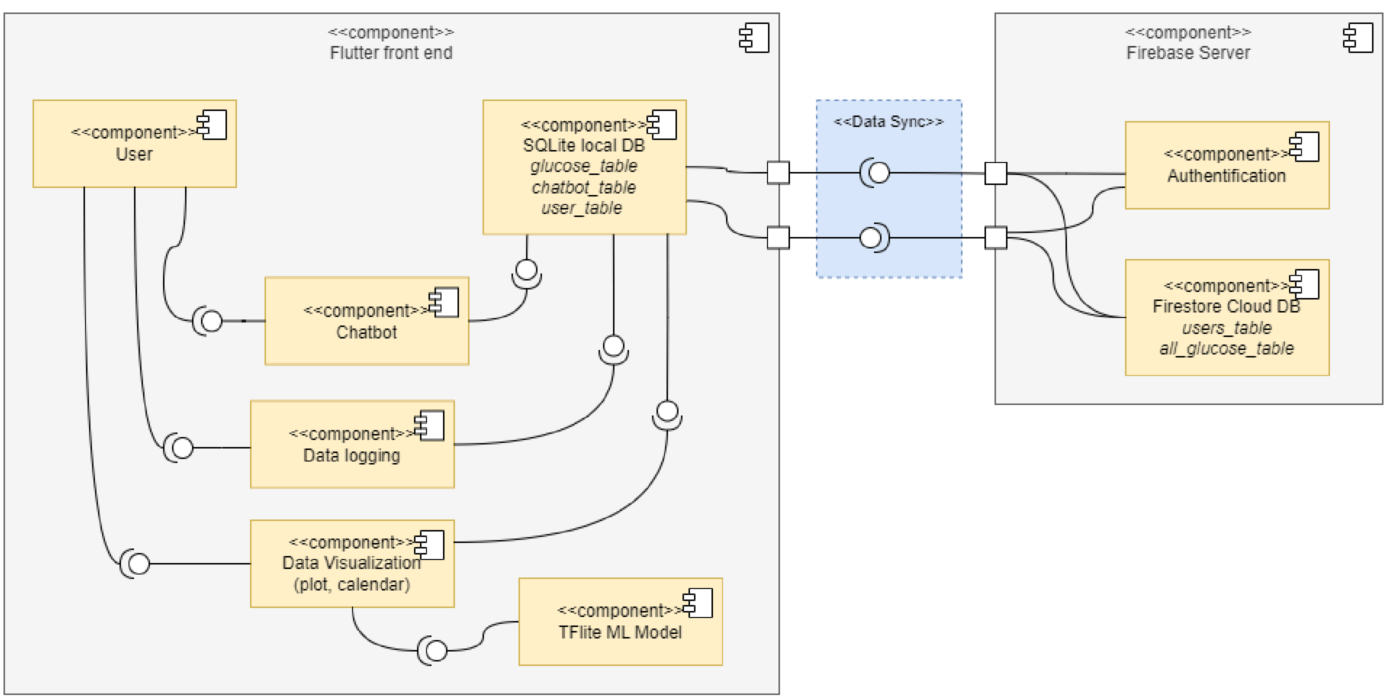
We deployed methods used to tackle imbalanced data, which is upsampling, downsampling from scikit-learn (upsampling by repeating same profiles, upsampling by synthesizing new profiles, and downsampling by pruning similar profiles). However MLP and SVC trained on these altered balanced dataset still showed precision around 0.5 (which was no better than just a random assignment of classes).

By examining the dataset's protocol form and patient-level forms we found out that 45.5% of the patients were diagnosed with T1D at the age after 25 (103 over 226 patients for whom information was provided and whose CGM data was used in training), which is inconsistent with average age of T1D diagnosis of 9-13 years old (Islam, 2021). Furthermore, 30.5% (69/226) of patients in the training had ongoing diabetes complications, which made us conclude that participants were probably using CGMs prior to the study and more attentive to their diabetes management due to the present diabetes complications (e.g. diabetic neuropathy, retinopathy).

Another reason for such poor performance is absence of important features (physical activity and food intake). The reference studies had access to proprietary data with detailed information on food intake and reliable timestamps. As a result, for each profile with NH with a certain set of features there was an identical set of features with no NH, because in the first case the patient went to bed hungry, while in the second case he/she might have eaten before bed, but it wasn't recorded in the dataset. Consequently, we decided to abstain from further training and focus on other tasks of the project.

Mobile app

Mobile app was developed using Flutter framework on front end, Firebase on back end (authorization and Firestore cloud database) and SQLite for a local database. For design we used Figma.



[3] Link the additional dataset used https://public.jaeb.org/direcnet/stdy/159

Figure 2. Component design of the mobile app

In short, the app has the following features:

1. Email/password and Google authentication via Firebase authentication feature
2. Logging BG readings via drag & drop (CRUD implemented)
3. Data sync and backup (the app account can be used on multiple devices, while the data of BG readings will stay consistent, as the data is loaded to and downloaded from the Firestore NoSQL database)
4. Data visualization (plots and calendar indications)
5. Rule-based chatbot to monitor statistics

The features numbered at 4 and 5 appeared as we progressed with the app development, and spotted a potential excellent UX, as both were novel among existing diabetes apps and would embrace our goal of creating an app with high usability.

1. Drag & drop data logging

Most of the available diabetes apps have slow data logging with poor UX, despite the fact that it is the most used feature. As a solution we proposed a drag & drop method to replace slow typing of BG readings and the timestamp. To add BG a user will expand the colored scale, drag the needed BG and drop it on a time interval. Time intervals can be expanded or shrunk via finger swiping, while BG values can be deleted by long press and updated.

The issue with the design on Figure 6 was in limitations of Flutter. Specifically, curves were implemented by Path class, which didn't have a built-in Gesture Detector to allow dropping BG values onto it and swiping to change its shape. Furthermore, it was feasible to put Path under a stand-alone Gesture Detector, since it didn't accept it as a valid child. As a result we switched to developing a more straight-forward solution shown in Figure 7.

In the new solution the user should stroll the timeline up and down to access the different dates (the date is indicated in the bottom-right corner box). To add the BG value a user firstly taps on the upper scale (in simple terms, to choose the integer part of the BG to reveal the scale with decimal part) and then drags the the BG value and drops it on the timeline. Each time block stores time borders and BG values that belong to the time interval (if multiple values are present, but it shows the average). A time block can be expanded or shrunk by double tapping, which triggers creation of a new timeblock or deletion of the time block respectively.

By long-pressing the colored BG values on the time block a user can trigger a deletion process, which opens a red colored circle with the cross. If tapped the BG value will be deleted.Alternatively, if a user drags a new BG value over an existing one, then the old value will be updated. Ultimately, all the manipulations are served by CRUD.

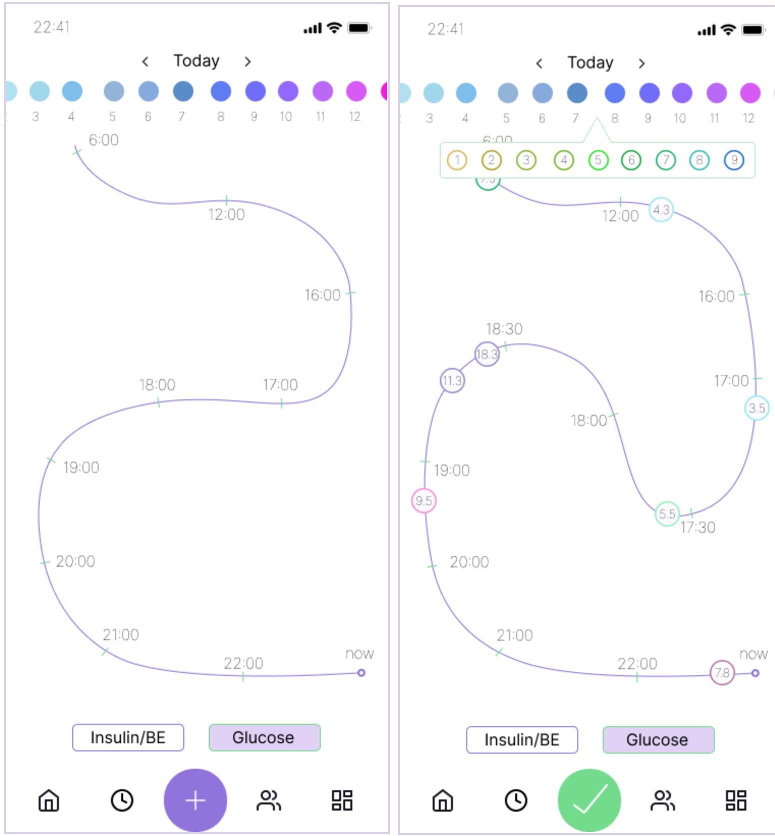


Figure 6. Initial design of drag&drop data logging (Figma design, not a real app screenshot)

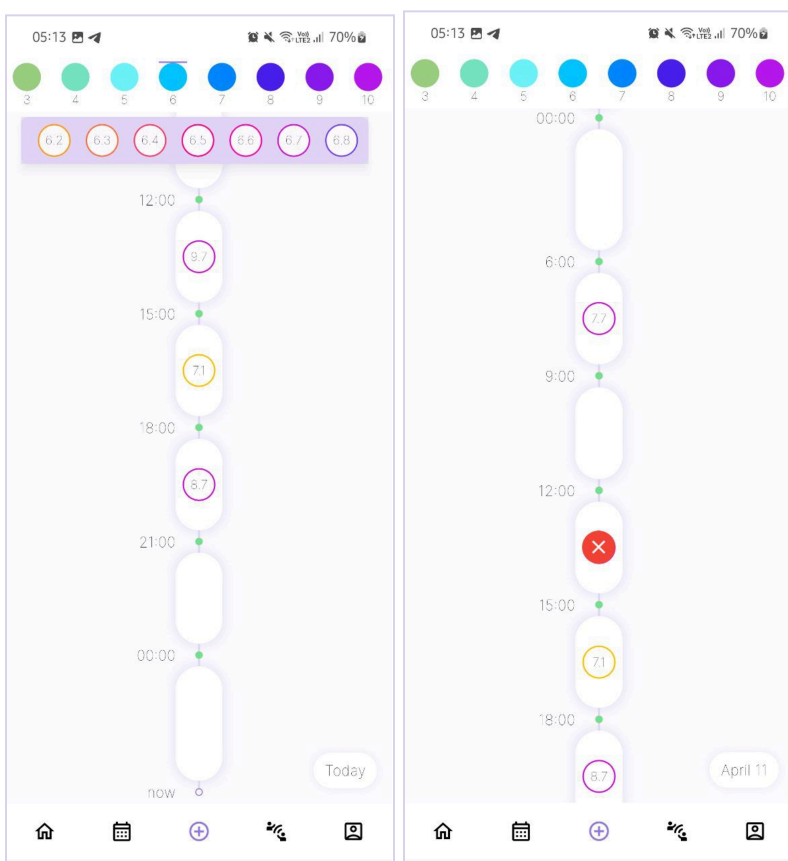


Figure 7. Implementation of drag & drop data logging (on the second frame the red cross indicates that the glucose value circle was long-pressed, which triggered deletion function)

2. ML data visualization

Most diabetes apps generate scatter plots based on BG entered throughout the day. However, these plots are unrealistic since they are based on just the data points provided by the user. The solution is to train a ML model to generate realistic blood fluctuations throughout the day given the same number of data points. Ultimately, it will allow users to see how naturally the concentration of BGs decreases after spikes over time and that even while not eating or exercising it fluctuates.

For the dataset we leveraged on the one we used for NH prediction. The dataset included CGM data accompanied with intermittent BG (IBG) patients made throughout the day (3-5 on average). Consequently, CGM data were the target set (108 BG readings over 6:00 to 23:59), while IBG were the feature set (108 data points over 6:00 to 23:59 with 10 minute step all zeros except for the existing IBG), as shown on Figure 3. The data processing resulted in 5K instances. For the model evaluation we needed to test how well it replicates the IBG from the feature list (e.g. 7.7 mmol/L at 15:23 and 6.4 mmol/L at 19:50), while replacing zeros with predicted fluctuations of BG. As a result we applied RMSE and MAE over non-zero IBG.

For the model we chose CNN from TensorFlow. After tuning the hyperparameters (number of epochs, number of hidden layers, numbers of neurons in each layer and kernel sizes), the following configuration was set.

The trained model had RMSE of 0.65 and MAE of 0.51. Interestingly, RMSE and MAE of true CGM were 0.71 and 0.55 respectively, i.e. readings from CGM devices did not perfectly match the ones taken using glucometer (it can also be seen from Figure 4). As a result, our model outperformed the true values, matching more closely to IBG readings.

Consequently, we incorporated the trained model to the Flutter app using *tflite_flutter* packet. The result can be seen in Figure 5.

```
model = Sequential([
    Input(shape=(108, 1)),
    Conv1D(128, kernel_size=(8),
                 activation='relu',
                 padding='same'),
    Conv1D(64, (4), activation='relu', padding='same'),
    MaxPooling1D(pool_size=(2)),
    Flatten(),
    Dense(108)
])

model.compile(loss=keras.losses.mean_absolute_error,
              optimizer=keras.optimizers.Adam(learning_rate=0.001),
              metrics=keras.metrics.mean_absolute_error)
epochs = 50
model.fit(X_train, y_train, epochs=epochs, verbose=2, validation_split=0.1)
```

Figure 3. Configuration of the model

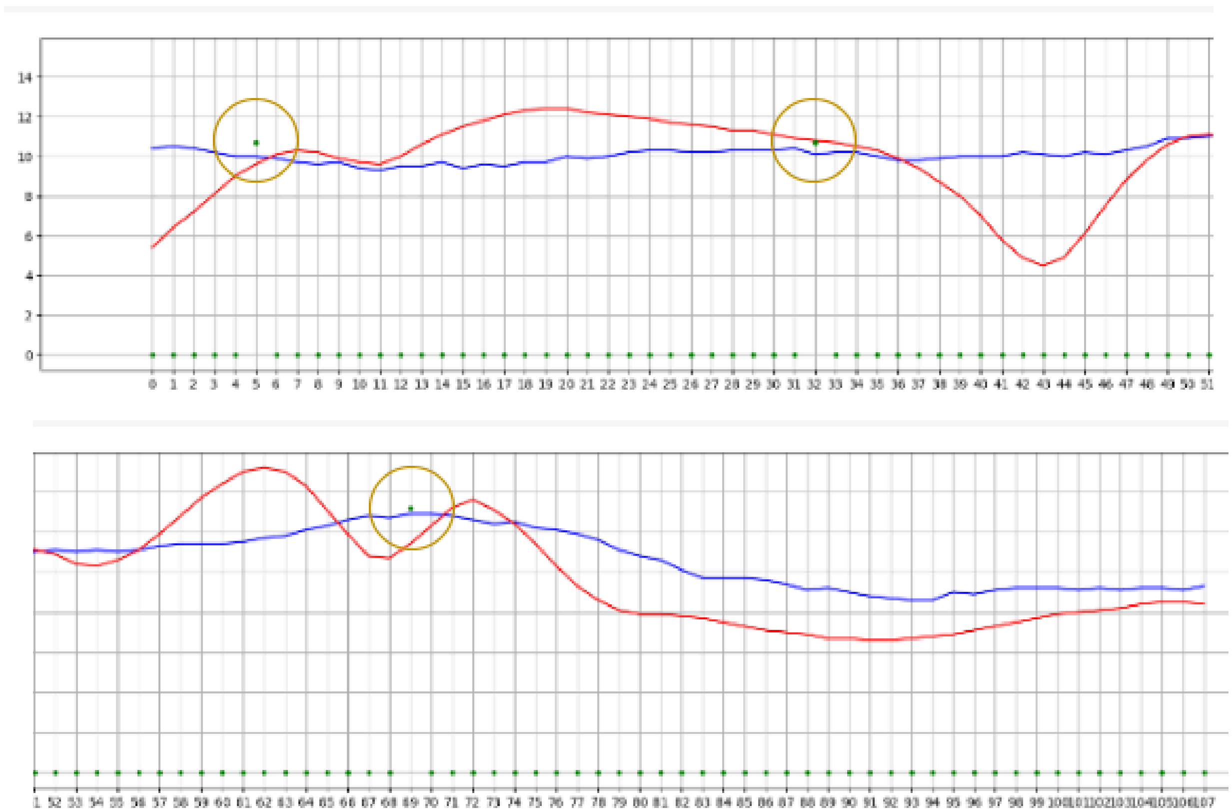


Figure 4. Green dots are the target set of IBG (non-zero values stand for existent BG readings that a patient made and were indicated by yellow circle); red line – true CGM; blue line – predicted CGM (Oy – BG values, Ox – time between 06:00 to 23:50 with 10 minute step represented numerically). The plot was split into two for the ease of viewing

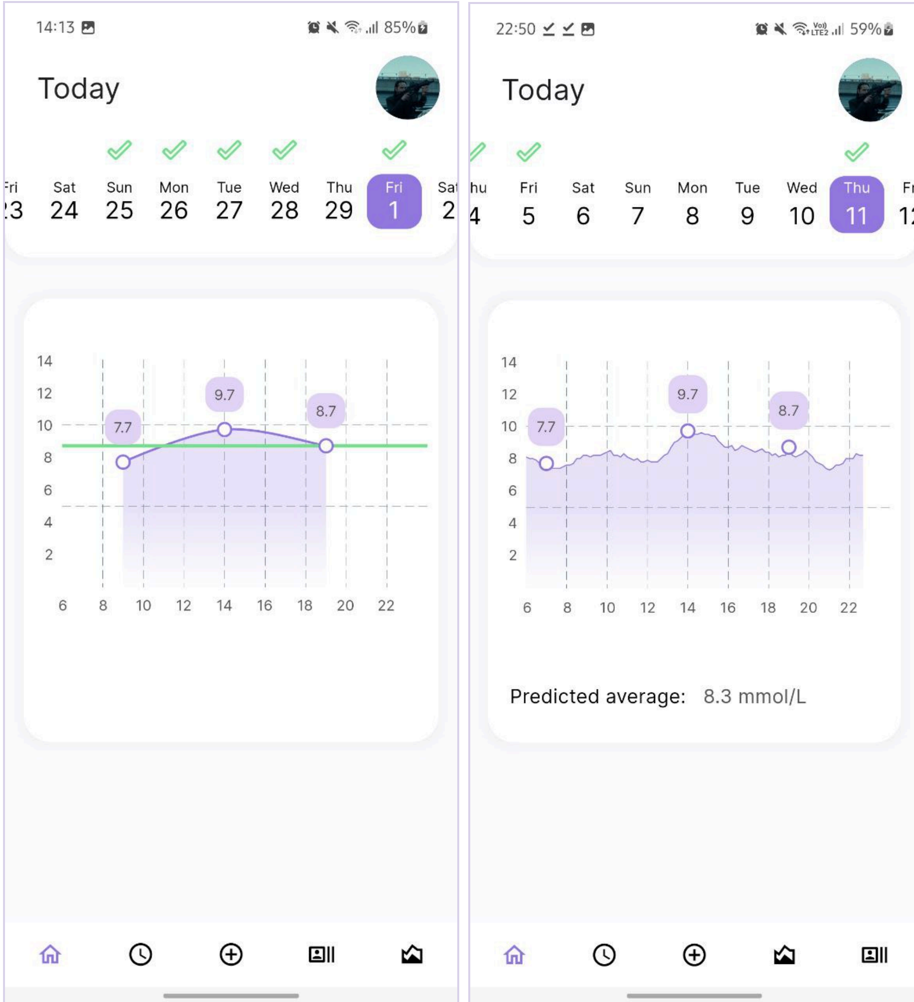


Figure 5. The same feature implemented in Fall'23 and the final version with ML (the plot extends further to fit to the current time (it was 22:50))

3. Rule-based chatbot

Most diabetes apps use visuals like pie charts, plots, and histograms.to display long term statistics. Instead, we can incorporate chatbot to generate SQL requests to the database based on user queries (e.g. for query “What is my average sugar for past 2 weeks” generate “SELECT AVG(blood_glucose) FROM glucose_readings WHERE reading_date >= DATE_SUB (CURDATE (), INTERVAL 2 WEEK);”) by training sequence-to-sequence LLMs like BART. However, we were unable to find suitable datasets with included time interval queries. As a result we switched to developing a rule-based chatbot. On Figure 5 it can be seen that we firstly choose the type of average BG (morning/afternoon/evening/night or 24 hours) and then select a time interval.

To distinguish the messages added coloring of the responses based on the BG value.

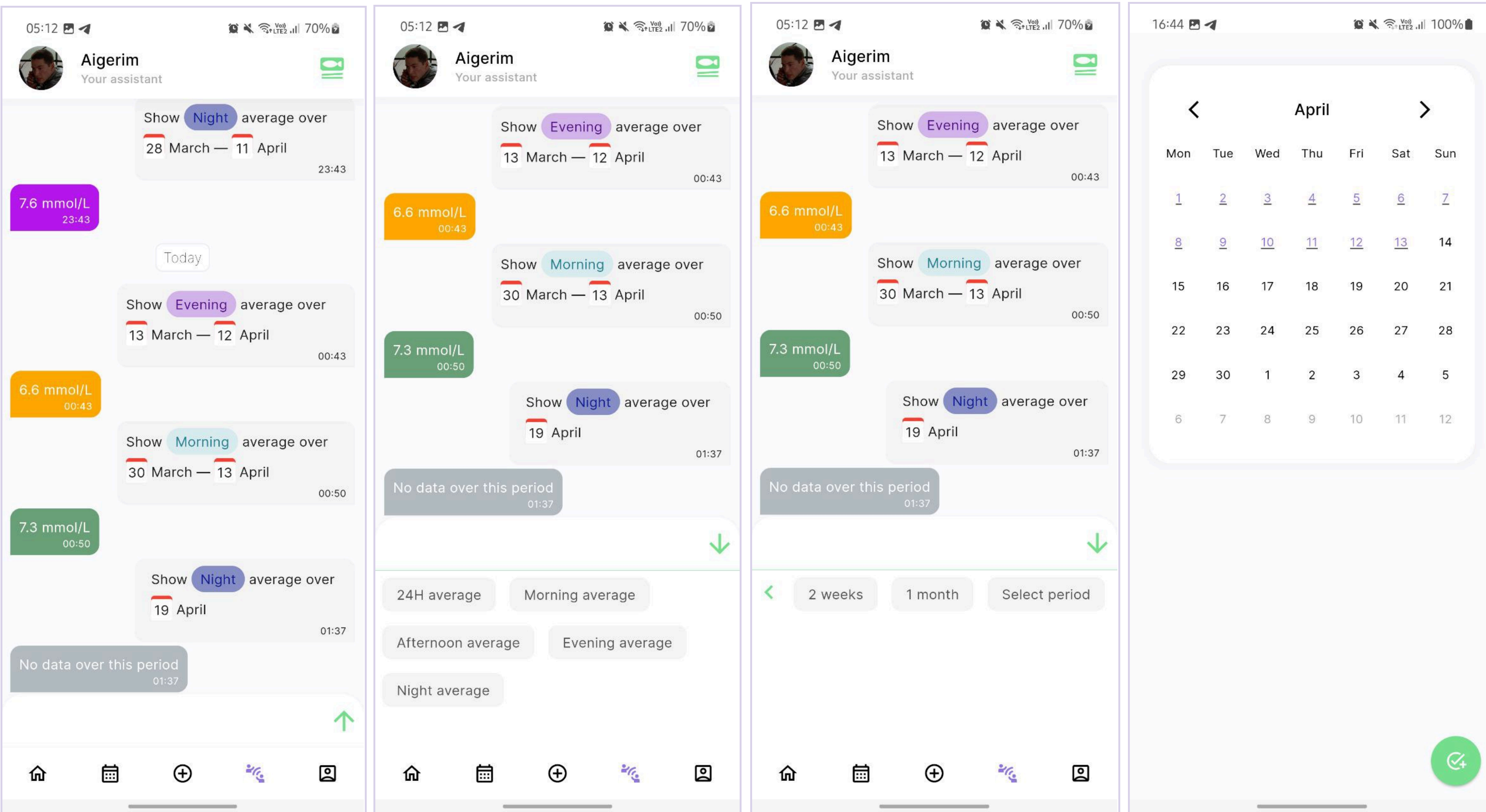


Figure 5. Implementation of rule-based chatbot (with “Select period” interface in the last frame)

4. Evaluation

The remaining UI of the app is shown in the Appendix A.

To evaluate the app we created a questionnaire (see Appendix B). We collected a response from a person diagnosed with T1D (one of the team members' younger brother who is 19 years old). He rated his overall satisfaction with the app at 4 / 5, while he rated efficiency and usability of the ML Plot, Data Logging, and Chatbot features at 5 / 5, 4 / 5, and 5 / 5. The issues that bothered him while using the app were bulky transitions and not intuitive design. In his long response he elaborated that in the data logging the while he swiped up and down the timeline the interface freezed for a slip of a second, and then went back to normal. Upon our investigation it turned out that this bug was due to the limitations of the flutter’s Infinite ListView widget, which stored only 3 elements at a time (the one that was visible, the upper and the bottom ones). The not intuitive design, as the respondent explained, was due to his initial struggle with understanding the Data Logging functionality (expanding and shrinking of the time blocks was not self-explanatory). Finally, we indicated a customization of design of the missing feature of the app, since he didn’t like the lavender color (the primary color of UI).

Overall, to address the issues in future we will add dynamic guidelines and hints, to teach users how to use the app. Additionally, we will add color customization in the settings.

Conclusion and possible future work

In conclusion, we developed a cross-platform mobile application ready to be deployed by people with diabetes, which incorporates novel for the diabetes apps functionalities: drag & drop data logging, ML data visualization, and rule-based chatbot for progress monitoring. We did not succeed in training a Nocturnal Hypoglycemia classifier to up to a good performance due to a dataset lacking data on patients' food intake. However, using the same dataset we were able to train a CNN model to generate BG fluctuations plots over 06:00 to 23:59 time interval given just a few of intermittent BG values with RMSE of 0.65 and MAE of 0.51.

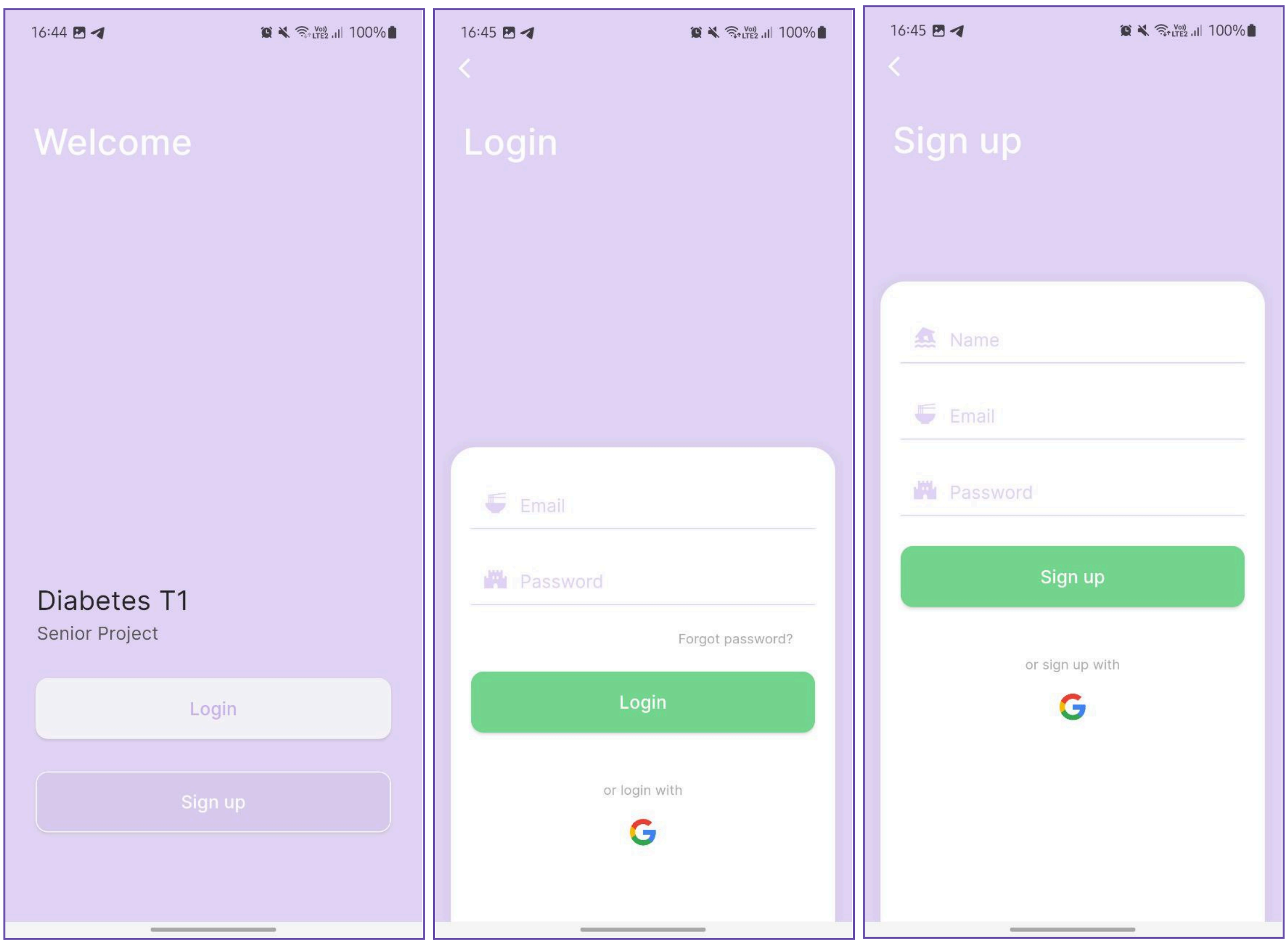
The future work might be testing the app in a professional environment, fixing the bug and uploading the app to App Store and Google Play for a user to use on a free basis.

Link to source codes

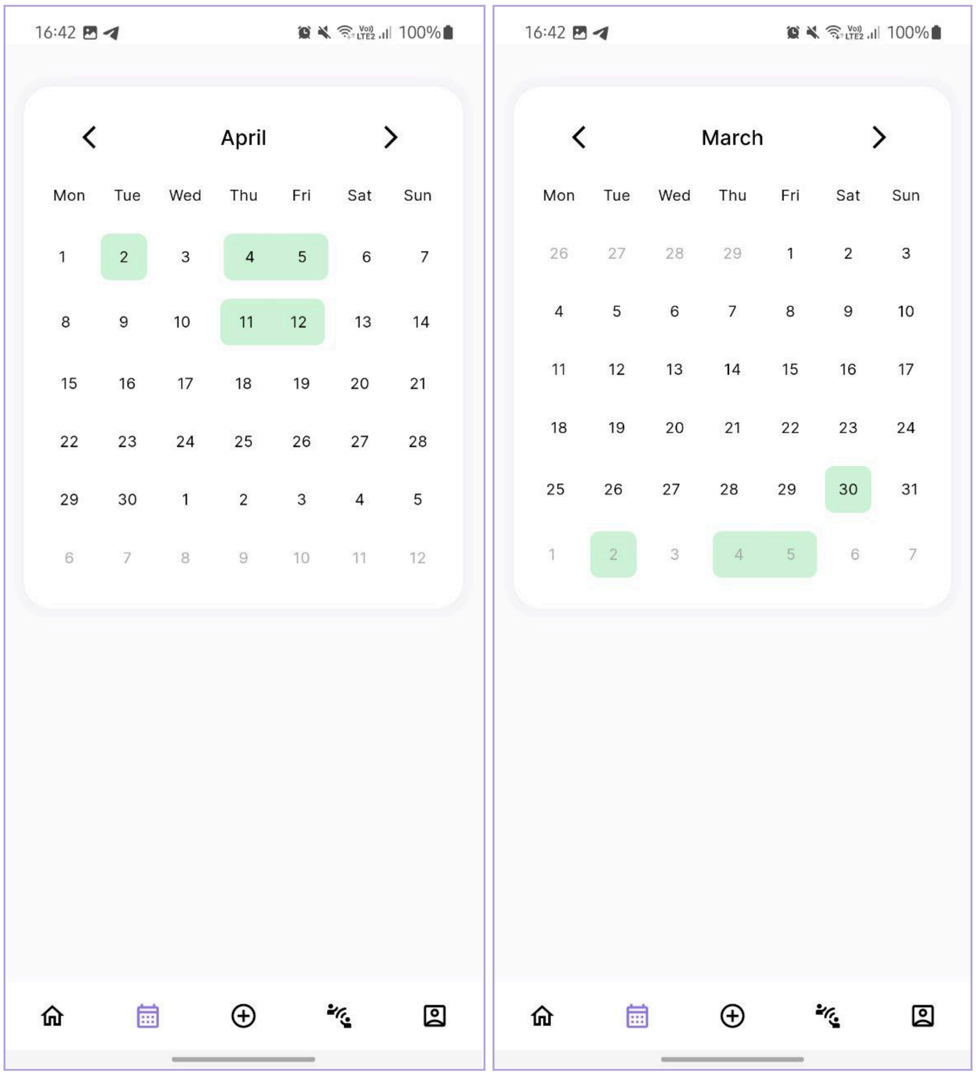
Git repository of the mobile app:
https://github.com/liebesbachlein/t1d-app.git (see main branch: ***aigerim-branch***)

Git repository containing .ipynb files of ML data processing, model training, evaluation and datasets used (original and processed):
https://github.com/liebesbachlein/ml_t1d.git

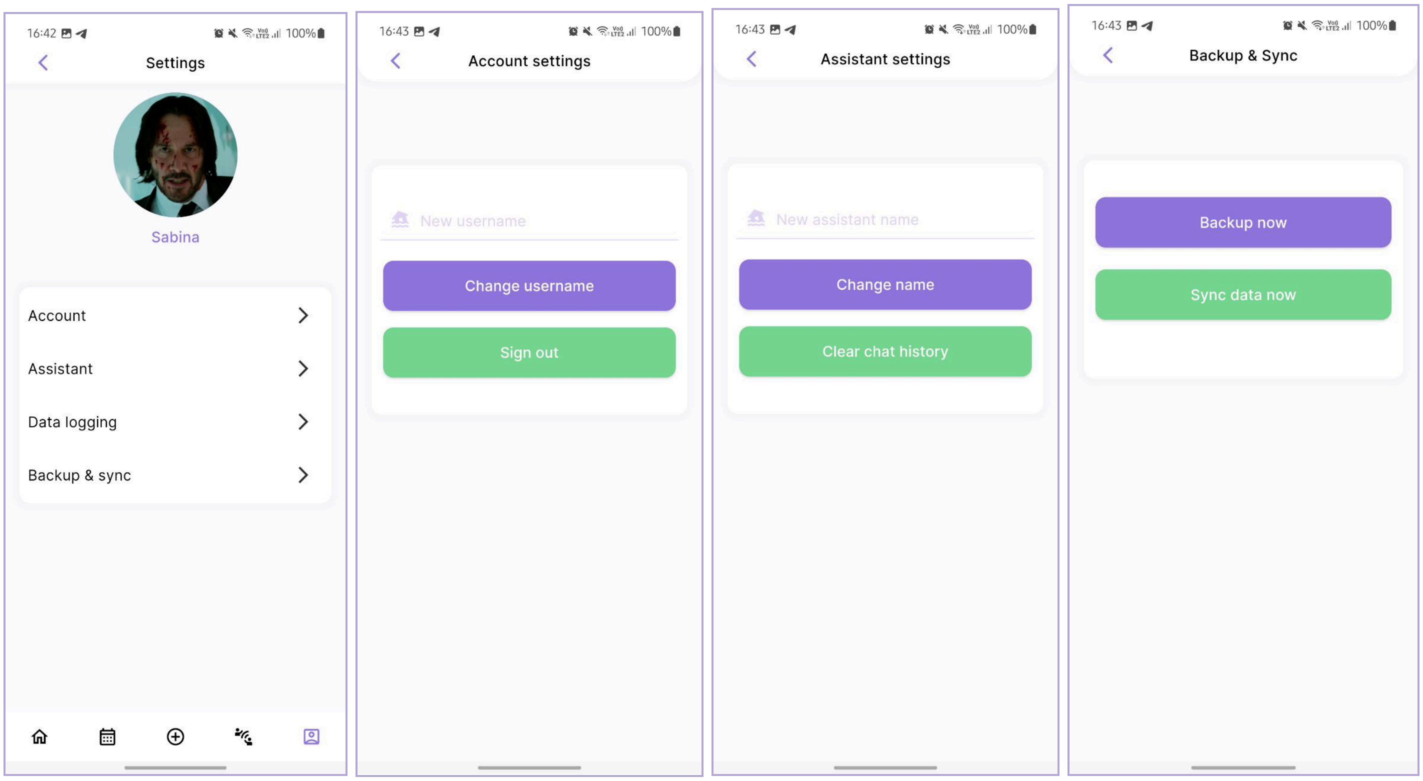
Appendix A



Authorization UI



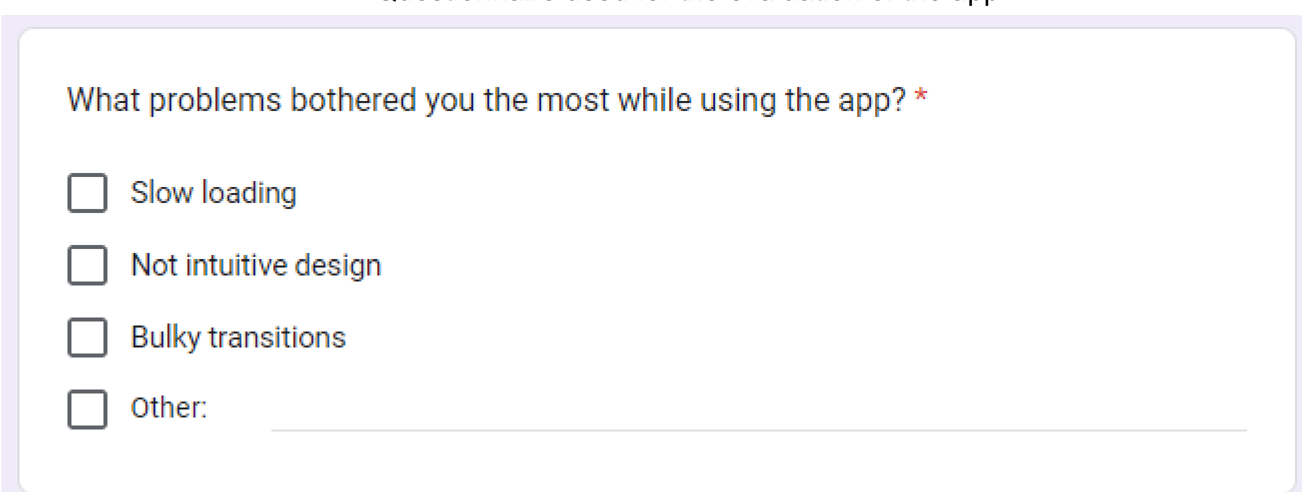
UI of the Calendar Page, where green color indicates that a user logged data for that day (by clicking the arrow a user can navigate between months)



UI of the Settings Page (Data Logging settings were removed. They included customization of the pre-set time intervals introduced bugs and overall were redundant)

Appendix B

Questionnaire used for the evaluation of the app



What problems bothered you the most while using the app? *

- [ ] Slow loading
- [ ] Not intuitive design
- [ ] Bulky transitions
- [ ] Other: ____________

Elaborate on the issues you encountered in detail *

Your answer

Rate the efficiency of the data logging feature *

	1	2	3	4	5	
Bad, unusable	○	○	○	○	○	Excellent, effective

Rate the efficiency of the chatbot feature *

	1	2	3	4	5	
Bad, unusable	○	○	○	○	○	Excellent, effective

Rate the efficiency of the plot feature *

	1	2	3	4	5	
Bad, unusable	○	○	○	○	○	Excellent, effective

Rate the your overall satisfaction with the app *

1 2 3 4 5

What feature did you expect but not find?

Your answer

References

1. Parcerisas, A., Contreras, I., Delecourt, A., Bertachi, A., Beneyto, A., Conget, I., Viñals, C., Giménez, M., Vehi, J. A Machine Learning Approach to Minimize Nocturnal Hypoglycemic Events in Type Patients under Multiple Doses of Insulin. *Sensors*, vol. 22, 2022. https://doi.org/10.3390/s22041665

2. M. Arnhold, M. Quade, and W. Kirch, "Mobile Applications for Diabetics: A Systematic Review and Expert-Based Usability Evaluation Considering the Special Requirements of Diabetes Patients Aged 50 Years or Older," *Journal of Medical Internet Research*, vol. 16, 2019. https://doi:10.2196/jmir.2968.

3. M. Kirwan, C. Vandelanotte, A. Fenning, M. J. Duncan, "Diabetes Self Management Smartphone Application for Adults With Type 1 Diabetes: Randomized Controlled Trial," *Journal of Medical Internet Research*, vol. 15, 2019. https://doi:10.2196/jmir.2588

4. A. Bertachi, C. Viñals, L. Biagi, I. Contreras, J. Vehí, I. Conget, M. Giménez. Prediction of Nocturnal Hypoglycemia in Adults with Type 1 Diabetes under Multiple Daily Injections Using Continuous Glucose Monitoring and Physical Activity Monitor. *Sensors* vol. 20, 2020. https://doi:10.3390/s20061705

5. H. J. Morten, C. Dethlefsen, P. Vestergaard, O. Hejlesen. Prediction of Nocturnal Hypoglycemia From Continuous Glucose Monitoring Data in People With Type 1 Diabetes: A Proof-of-Concept Study. *Journal of Diabetes Science and Technology*, vol. 14, 2020. doi: 10.1177/1932296819868727

6. P. Calhoun, R. A. Levine, J. Fan. Repeated Measures Random Forests (RMRF): Identifying factors associated with nocturnal hypoglycemia. *Biometrics,* vol. 11, 2020. doi: 10.1111/biom.13284

7. S. Islam. Diabetes: from Research to Clinical Practice. *Springer,* 2021