

Bæredygtig Energi fra Solceller

Tværfagligt projekt DMU D24 Sønderborg



Dato for aflevering: 29. november 2024

Indholdsfortegnelse

| | |
|---------------------------------|----|
| Indledning | 2 |
| Liste med tasks..... | 3 |
| Processen | 4 |
| Udarbejdelse af tasks | 4 |
| Brugergrænsefladens layout..... | 4 |
| Bæredygtigt design | 5 |
| Screendumps | 6 |
| Vigtige kodestumper..... | 6 |
| Den kørende applikation..... | 9 |
| Link til Github | 12 |
| Konklusion..... | 13 |
| Bilag..... | 14 |

Indledning

Dette projekt har til formål at udvikle en JavaFX applikation, der kan hjælpe en medarbejder med at visualisere energiproduktionen fra solcelleanlæg i en dansk kommune.

Data til projektet er fundet via følgende link:

<https://www.opendata.dk/city-of-aarhus/solcelleanlaeg>

Projektet kombinerer programmering, systemudvikling og grafisk datavisualisering, hvor målet er at levere et brugervenligt værktøj, der kan illustrere produktionen over tid, gennem grafiske diagrammer og statistiske beregninger.

Gruppen har, i løbet af projektperioden, arbejdet tæt sammen for at skabe en løsning, der opfylder kravene til både teknisk funktionalitet og brugervenlighed.

Opgavens data, der er baseret på et datasæt fra Open Data Aarhus, giver muligheden for at analysere energiproduktionen fra solcelleanlæg. Dette danner grundlaget for applikationens funktioner, som inkluderer:

- Indlæsning og behandling af "rå" data fra filen i en objektmodel.
- Mulighed for at vælge specifikke datoer og diagramtyper.
- Grafisk fremstilling af energiproduktionen time for time samt den samlede produktion for en specifik måned.

I rapporten dokumenteres processen og gruppens løbende overvejelser samt code-snippets af vigtige kodelumper og den endelige applikation. Rapporten indeholder også linket til gruppens GitHub-repository samt en liste med tidsestimeringer på opgavens tasks.

Liste med tasks

| Task | Estimeret tid | Reelt tidsforbrug |
|-------------------------|---------------|-------------------|
| Indlæs fil/data | 4 timer | 1 ½ time |
| Vis data på diagram | 3 timer | 2 timer |
| Vælg site | 1 time | 10 min |
| Vælg dato | 1 time | 2 timer |
| Vælg tidspunkt på dagen | 1 time | 2 timer |
| Vælg måned og år | 2 timer | 3 timer |
| Vælg diagramtype | 3 timer | 1 time |
| Lav design/UI | 8 timer | 10 timer |

Processen

For at skabe et overblik over vores projekt, udarbejdede vi tidligt i processen en papirprototype af hvordan vores brugergrænseflade kunne se ud (Bilag 1). Denne tilgang hjalp os med at visualisere designet og strukturere de funktionelle elementer i applikationen.

Udarbejdelse af tasks

For at skabe overblik og sikre en struktureret tilgang til projektet, udarbejdede vi en task-liste baseret på opgavens kravspecifikation og de mål, vi skulle opnå. Denne liste hjalp os med at bryde projektet ned i mindre dele, som kunne prioriteres og tidsestimere.

Arbejdet med tasks startede med en gennemlæsning af opgaveteksten og en fælles brainstorm i gruppen. Her identificerede vi de vigtigste opgaver, såsom indlæsning af data, implementering af statistikberegninger, og design af brugergrænsefladen. Vi brugte denne brainstorm-session til at oprette en række overordnede kategorier, som senere blev opdelt i mere specifikke tasks. Under udarbejdelsen af task-listen, prioriterede vi opgaverne efter deres betydning for projektets funktionalitet for derefter at identificere afhængighederne mellem tasks, således at vi kunne arbejde i en logisk rækkefølge. Efterfølgende har vi forsøgt at estimere tidsforbruget for hver task og slutteligt noterede vi det faktiske tidsforbrug, så vi kunne sammenligne og lære af eventuelle afvigelser.

Brugergrænsefladens layout

For at sikre en tydelig struktur i vores brugergrænseflade samt sikre en nem integration med andre layouttyper, valgte vi at anvende et BorderPane som grundlæggende layout. Dette layout giver mulighed for at organisere brugerfladens elementer i regioner (top, bund, venstre, højre og center), hvilket gør det velegnet til vores projekt.

For at skabe et struktureret og overskueligt layout, har vi i toppen af vores BorderPane, placeret et GridPane der indeholder elementer til interaktion. GridPane giver mulighed for at placere elementerne i et gittermønster, hvilket sikrer en ensartet og pæn organisering af labels, input felter og knapper. Her er det muligt for brugeren at vælge et "site" via en dropdown samt angive "Date". Derudover ses en textfield som viser hvilken Chart type der vises. Desuden er tilføjet en Button kaldet "Search", som brugeren kan klikke på, for at hente og vise den valgte data.

I centeret af BorderLayout har vi anvendt et StackPane til visning af diagrammer. Valget af StackPane er gjort, da denne type Pane placerer sine komponenter ovenpå hinanden, hvilket gør det muligt at skifte mellem visning af forskellige diagrammer hvor kun det øverste element vil være synligt. Derudover placerer StackPane vores indhold i centeret som standard, hvilket giver et pænt layout. Denne struktur gør det nemt for brugeren at navigere mellem forskellige datasæt og diagramtyper, samtidig med at layoutet forbliver overskueligt og brugervenligt.

I bunden af vores BorderLayout, har vi placeret en VBox, hvori der er placeret to labels. Øverste label viser overskriften "Total Production in kWh" og nederste label viser den statistiske beregning. I vores VBox er placeret en GridPane, for at indsætte to centrerede buttons kaldet "Day" og "Month". Formålet med disse er at kunne vælge om man vil se grafen og den pågældende dag, herunder time for time, eller om man vil se energiproduktionen for hele den pågældende måned.

De ovenstående layouttyper er udvalgt for at sikre fleksibilitet, brugervenlighed og en tydelig opdeling af funktionerne i brugergrænsefladen.

Bæredygtigt design

Designet i vores UI er nøje udvalgt til at afspejle bæredygtighedens kerneværdier. Farvepaletten og det overordnede layout, er skabt med inspiration fra naturen og miljøet, hvor de afdæmpede grønne og blå toner minder om naturens egne farver. Dette sikrer ikke blot et æstetisk tiltalende design, men også en visuel repræsentation af projektets overordnede emne, der omhandler bæredygtighed.

Screen.dumps

Vigtige kodestumper

```
10 public class ReadData  ▶ hund555 +1
11 {
12     /**
13      * Reads a file located at a given path to create SolarData objects
14      * witch are then returned as a list
15      * @param filepath Path to the file to read
16      * @return ArrayList of SolarData to be used in program
17      * @throws FileNotFoundException if the given path does not exists
18      */
19 @ public static ArrayList<SolarData> readFileData(String filepath) throws FileNotFoundException 1 usage ▶ sim
20 {
21     int id = 0;
22     String date = "";
23     String time = "";
24     int siteID = 0;
25     int totalEnergyProduction = 0;
26     int online = 0;
27
28     DateTimeFormatter formatting = DateTimeFormatter.ofPattern("yyyy-MM-dd"); // set formatting for date
29     ArrayList<SolarData> dataset = new ArrayList<>();
30
31     File inputFile = new File(filepath);
32     Scanner in = new Scanner(inputFile).useDelimiter("pattern: "[\t|T]"); // split by tabs and T
33
34     in.nextLine(); // skip first line (tsv table header)
35
36     while (in.hasNext())
37     {
38         // read data from tsv file
39         id = in.nextInt();
40         date = in.next();
41         time = in.next();
42         siteID = in.nextInt();
43         totalEnergyProduction = in.nextInt();
44         online = in.nextInt();
45
46         in.nextLine();
47
48         // convert date and time to the appropriate datatype and class
49         LocalDate dateConverted = LocalDate.parse(date, formatting);
50         int timeConverted = Integer.parseInt(time.substring(0, 2));
51
52         // add SolarData objects with arguments to arraylist
53         dataset.add(new SolarData(id, dateConverted, timeConverted, siteID, totalEnergyProduction, online));
54     }
55     return dataset;
56 }
57 }
```

Figur 1: En klasse der læser data fra filen, så vi kan bruge den.

```
39 // Initializes and reads the data file
40 @FXML TobiasFalck +1
41 public void initialize() {
42     try {
43         data = ReadData.readFileData(filepath: "src/resources/solar-dataset.tsv");
44
45         for (SolarData solarData : data) {
46             // finds all siteIDs that are not currently added to siteDDL
47             if (!siteDDL.getItems().contains(String.valueOf(solarData.getSiteID()))) {
48                 // convert int to String, since choicebox only takes Strings
49                 siteDDL.getItems().add(String.valueOf(solarData.getSiteID()));
50             }
51         }
52
53         // Ensure both charts start hidden
54         productionBarChart.setVisible(false);
55         productionLineChart.setVisible(false);
56
57     } catch (FileNotFoundException e) {
58         errorMessage.setText("File not found");
59     } catch (Exception e) {
60         errorMessage.setText(e.getMessage());
61     }
62 }
```

Figur 2: Læser vores datafil og finder alle sites-ID. Efterfølgende gør den vores diagrammer usynlige.


```

64 // The on action for the search button
65 public void createChartClick() { 1 usage 1 TobiasFalck +2 *
66     ArrayList<Integer> totalsWhs = new ArrayList<>();
67     ArrayList<Integer> times = new ArrayList<>();
68     HashMap<Integer, Integer> dailyTotals = new HashMap<>();
69
70     diagramTypeDDL.setText("Bar Chart");
71     errorMessage.setText(""); // hide error message
72     productionTotal.setText(""); // Resets the production total label
73
74     LocalDate datePicked = dateDP.getValue();
75     int siteIDPicked = Integer.parseInt(siteDDL.getValue()); // get siteID from sites choice box and convert it to int
76
77     for (SolarData solarData : data) {
78         // Check for the selected site and date match
79         if (siteIDPicked == solarData.getSiteID()) {
80             // Add to day chart data
81             if (datePicked.equals(solarData.getDate())) {
82                 totalsWhs.add(solarData.getWattPerHour());
83                 times.add(solarData.getTime());
84             }
85
86             // Add to month chart data
87             if (datePicked.getMonth() == solarData.getDate().getMonth())
88             {
89                 int day = solarData.getDate().getDayOfMonth();
90                 dailyTotals.put(day, dailyTotals.getOrDefault(day, defaultValue: 0) + solarData.getWattPerHour());
91             }
92         }
93     }
94
95     double totalProductionMonth = 0;
96     for (int dayTotal : dailyTotals.values()) {
97         totalProductionMonth += dayTotal;
98     }
99
100     productionTotal.setText(String.valueOf(d: totalProductionMonth / 1000) + " kWh"); // convert to String
101 }
102
103 // Generate charts if data is available
104 if (totalsWhs.isEmpty() && dailyTotals.isEmpty()) {
105     errorMessage.setText("No data for chosen date or month.");
106 } else {
107     createDayChart(siteIDPicked, datePicked, totalsWhs, times);
108     createMonthChart(siteIDPicked, datePicked, dailyTotals);
109 }

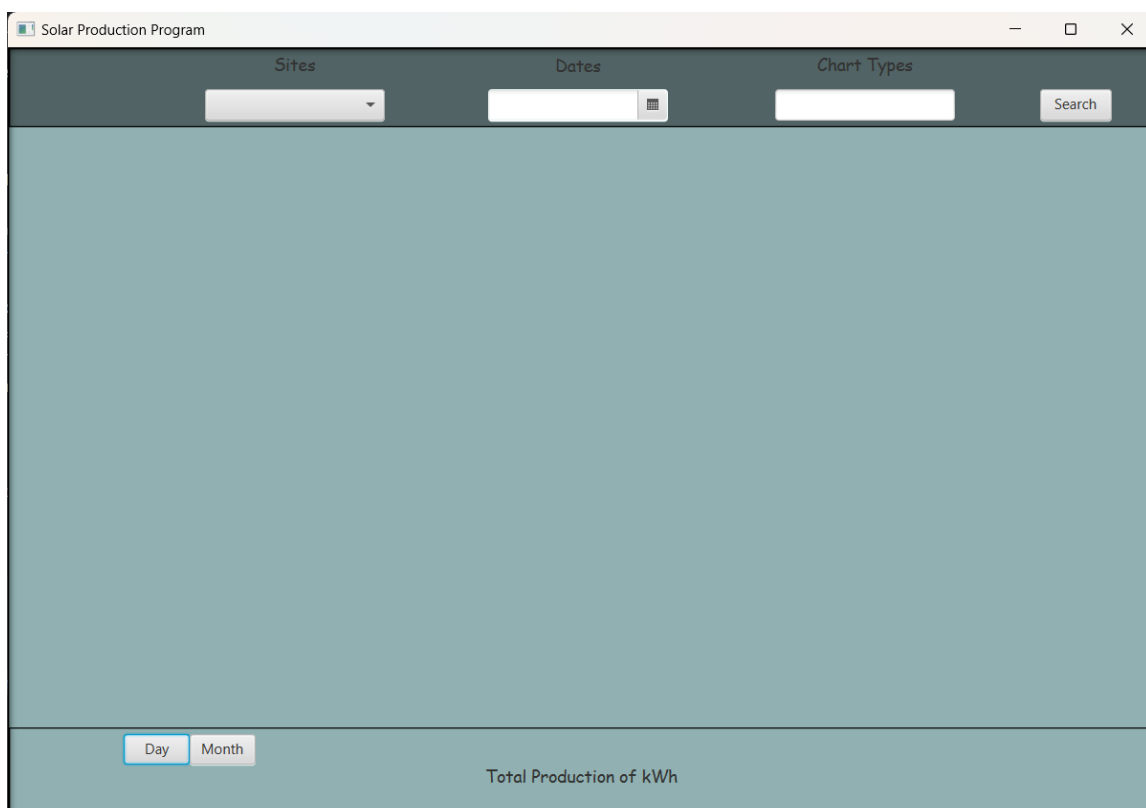
```

Figur 3: Vores knap som skaber en masse ArrayLister. Dernæst sætter den en default på vores Textfield og nulstiller både "errormessagen" og vores udregning. Så læser den data fra den valgte site og finder dato, måned og dag. Til sidst lægger den produktionen fra dagene sammen i en double og udregner den til kWh.

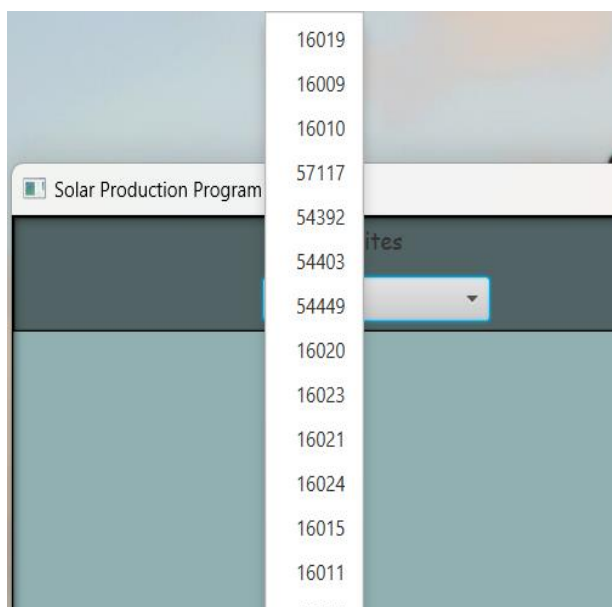
```
124 // Create the day chart
125 @ public void createDayChart(int siteIDPicked, LocalDate datePicked, ArrayList<Integer> totalWhs, ArrayList<Integer> times) {
126     XYChart.Series<String, Integer> series = new XYChart.Series<>();
127     series.setName("Site ID: " + siteIDPicked + "\nDate: " + datePicked.toString());
128
129     for (int i = 0; i < totalWhs.size(); i++) {
130         series.getData().add(new XYChart.Data<>("X: " + times.get(i) + ":00", totalWhs.get(i)));
131     }
132
133     productionBarChart.setData(FXCollections.observableArrayList(series));
134 }
135
136 // Create the month chart
137 @ public void createMonthChart(int siteIDPicked, LocalDate datePicked, HashMap<Integer, Integer> dailyTotals) {
138     XYChart.Series<String, Integer> series = new XYChart.Series<>();
139     series.setName("Site ID: " + siteIDPicked + "\nMonth: " + datePicked.getMonth());
140
141     for (int day : dailyTotals.keySet()) {
142         series.getData().add(new XYChart.Data<>("X: " + "Day " + day, dailyTotals.get(day)));
143     }
144
145     productionLineChart.setData(FXCollections.observableArrayList(series));
146 }
147 }
```

Figur 4: To metoder der skaber enten en "DayChart" eller en "MonthChart", hvor den får informationen fra ArrayLister.

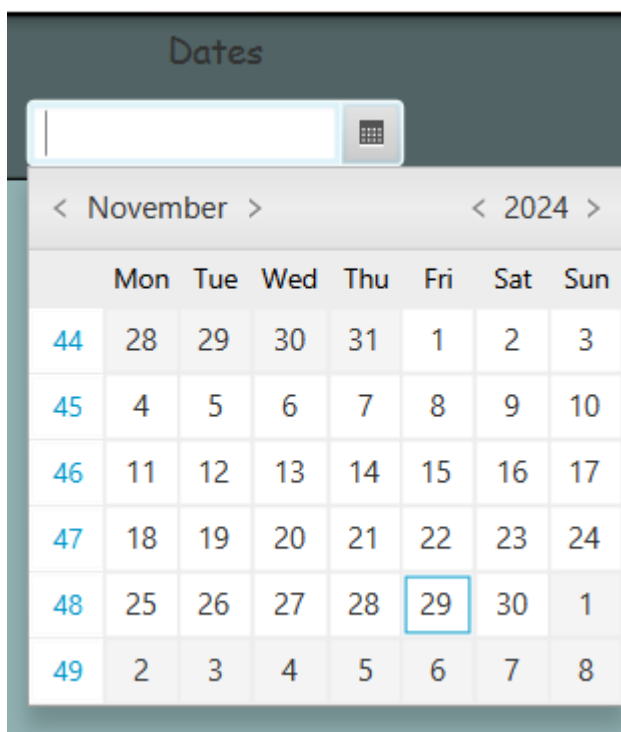
Den kørende applikation



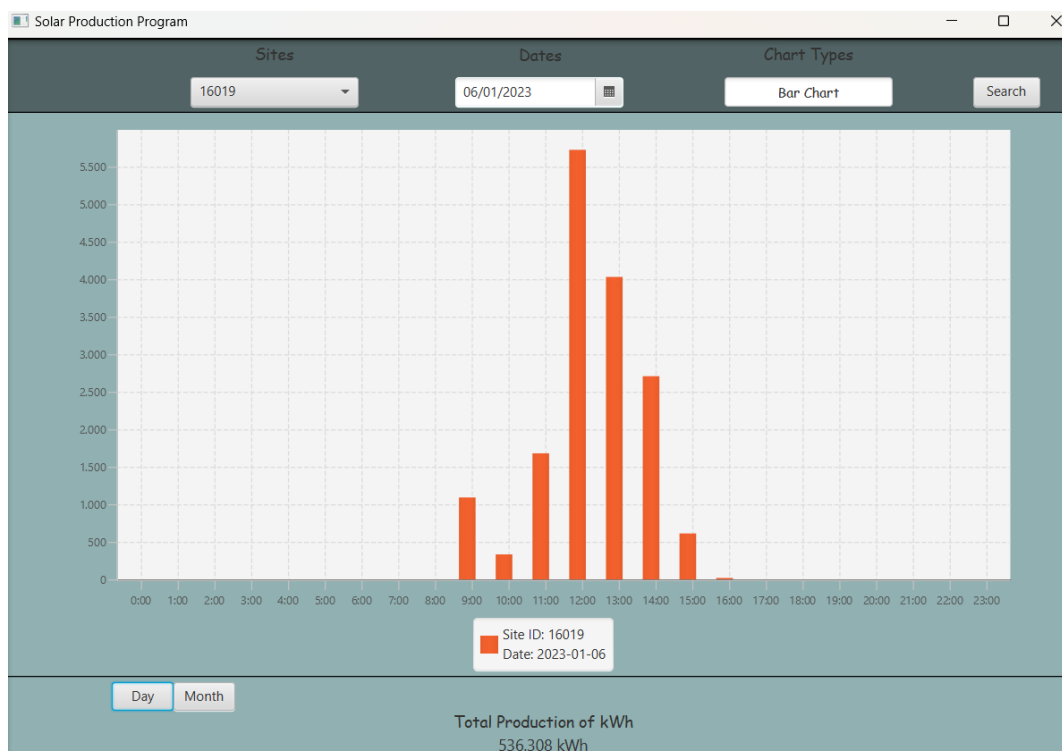
Figur 5: Billede af vores brugergrænseflade inden brugeren har gjort nogle valg.



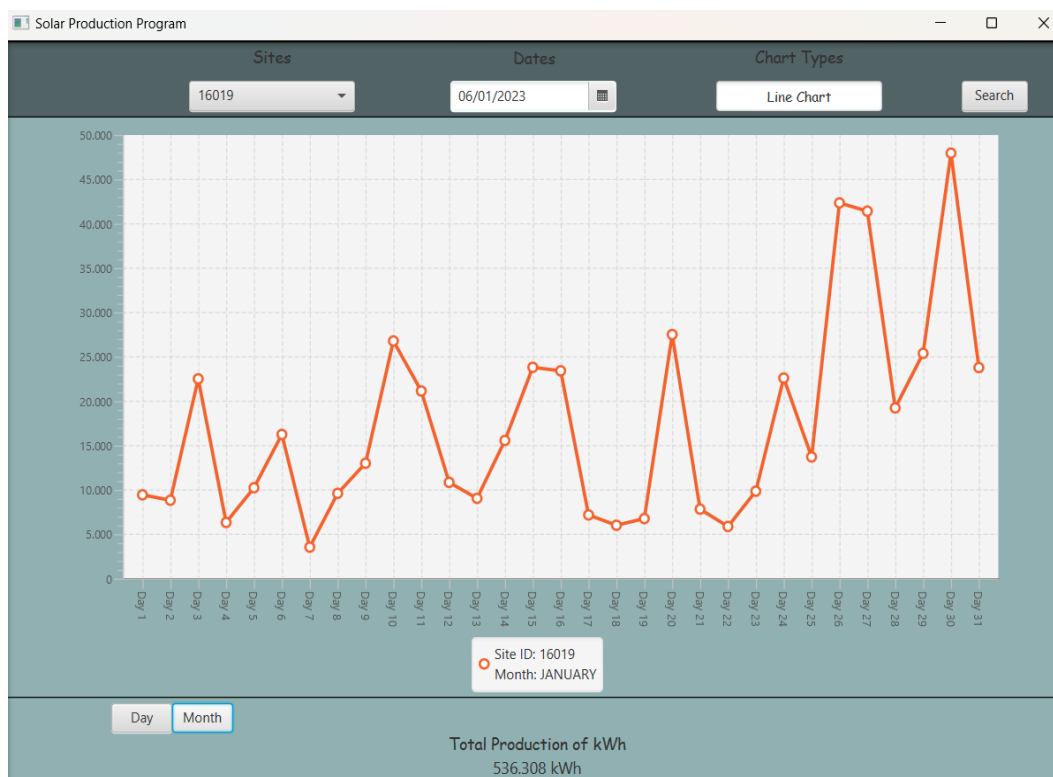
Figur 6: Dropdown menu for sites.



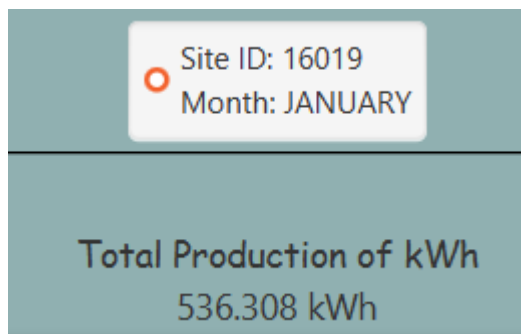
Figur 7: Vores datapicker for at vælge en dato.



Figur 8: Viser dag-graf for valgte dato.



Figur 9: Viser månedsgraf for valgte måned.



Figur 10: Viser udregning af månedens produktion.

Link til Github

[TobiasFalck/SolarProductionProject: Uge 48 Projekt](#)

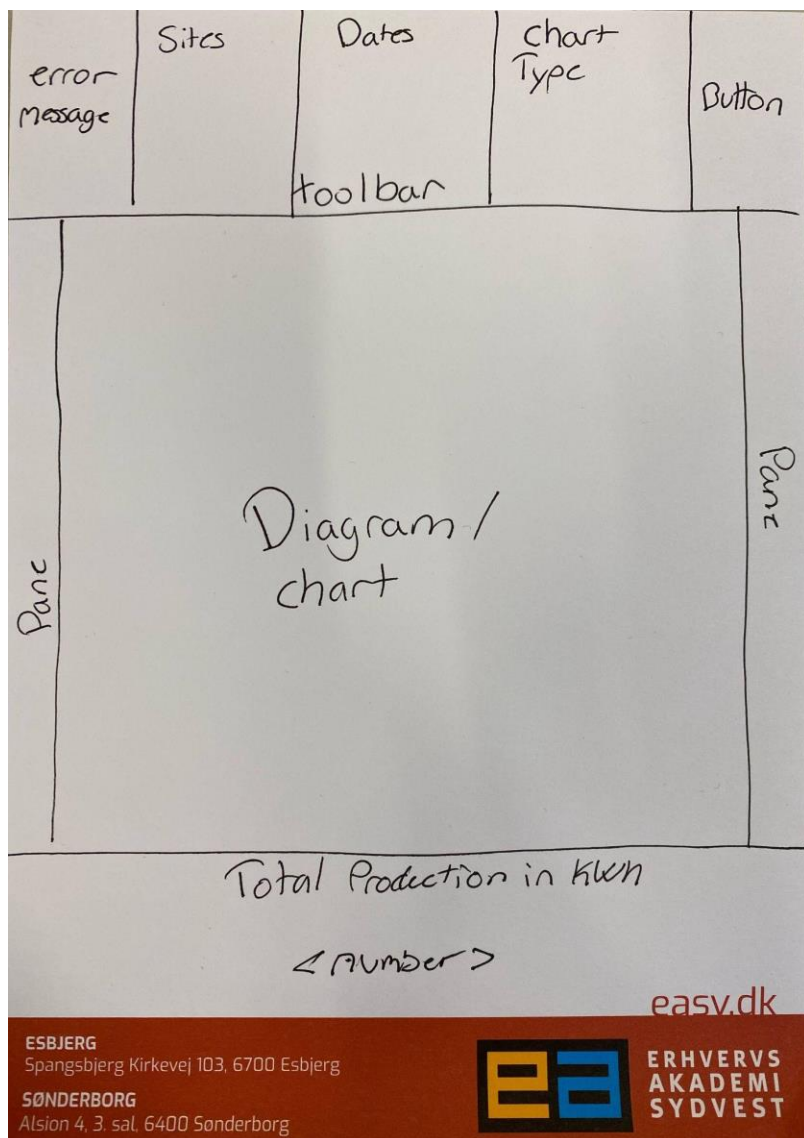
Konklusion

Projektet “Bæredygtig Energi fra Solceller” har været en lærerig og udfordrende proces, hvor vi som gruppe har arbejdet målrettet på at udvikle en brugervenlig JavaFX-applikation, der kan visualisere energiproduktion fra solcelleanlæg. Vi har opnået vores mål om at integrere datahåndtering, grafisk fremstilling og funktionelt UI-design i en applikation, der understøtter medarbejderens behov for overblik over energiproduktionen.

Gruppens samarbejde har fungeret godt, og vi har formået at udnytte vores forskellige kompetencer inden for programmering og systemudvikling effektivt. Tidsplanen og task-listen har hjulpet os med at strukturere arbejdet, selvom der var nogle opgaver, der krævede mere tid end forventet, fx valg af tidspunkt og dato. Selvom enkelte opgaver tog længere tid end planlagt, har projektet generelt været vellykket. Vi har opnået en større forståelse for konceptet MVC, filhåndtering i Java og udviklingen af grafiske applikationer.

Afslutningsvis kan vi konkludere, at vores applikation lever op til kravene i opgaven og kan fungere som et effektivt værktøj til at illustrere og analysere energiproduktionen fra solceller. Erfaringerne fra dette projekt vil vi tage med videre til fremtidige opgaver, hvor vi vil bygge videre på de metoder og strategier, vi har anvendt her.

Bilag



Bilag 1: Prototype af vores brugergrænseflade.