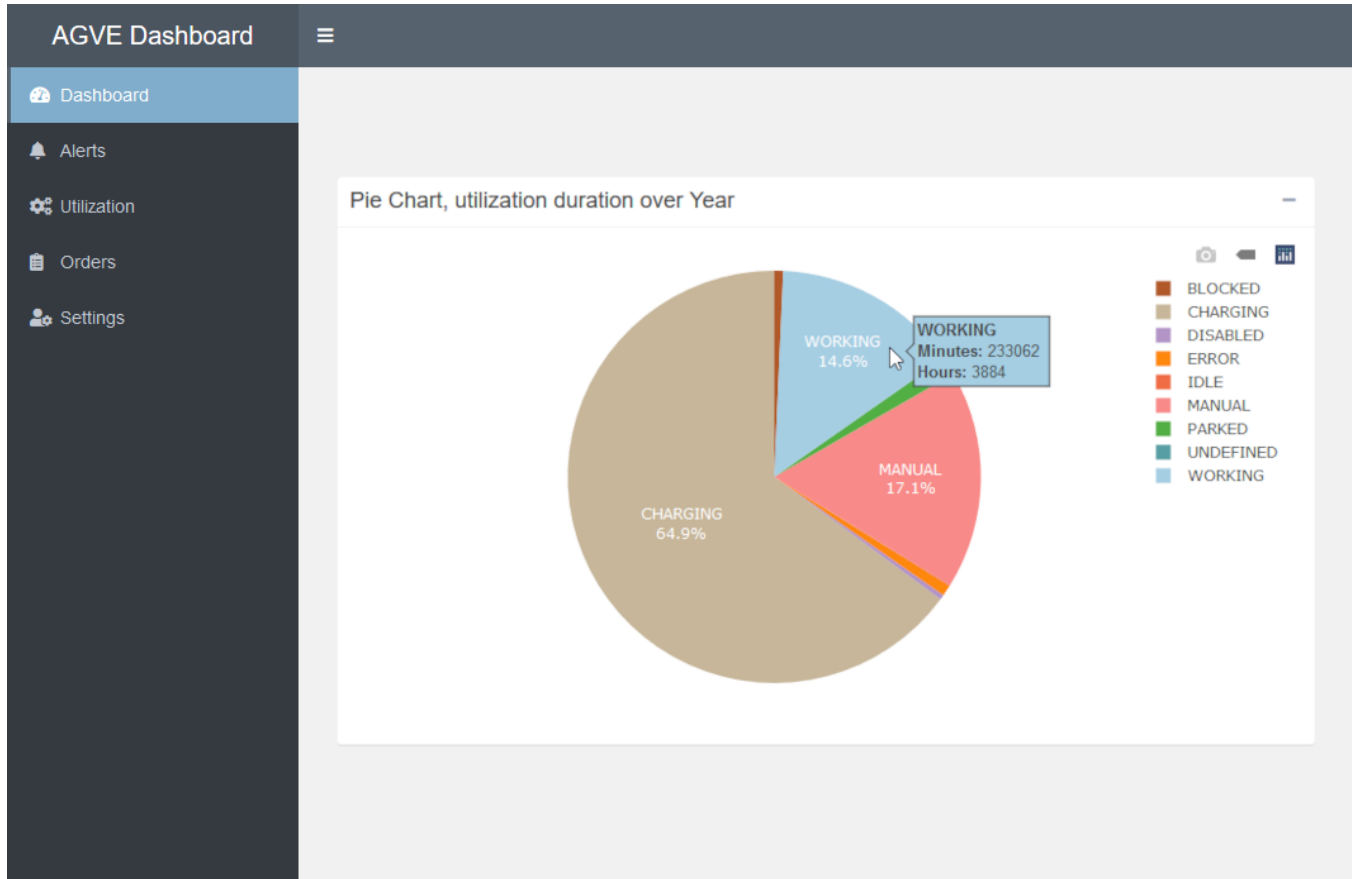




CHALMERS



Visualisering av AGV-statistik

Utveckling av en webb-dashboard i R

Examensarbete inom Data- och Informationsteknik

Matteus Straznyk
Adrian Håkansson

Institutionen för Data- och Informationsteknik
CHALMERS TEKNISKA HÖGSKOLA
GÖTEBORGS UNIVERSITET
Göteborg, Sverige 2021

EXAMENSARBETE

Visualisering av AGV-statistik

Utveckling av en webb-dashboard i R

Matteus Straznyk
Adrian Håkansson

Institutionen för Data- och Informationsteknik
CHALMERS TEKNISKA HÖGSKOLA
GÖTEBORGS UNIVERSITET

Göteborg 2021

Visualisering av AGV-statistik

Utveckling av en webb-dashboard i R

Matteus Straznyk

Adrian Håkansson

© Matteus Straznyk, Adrian Håkansson, 2021

Examinator: Arne Linde

Institutionen för Data- och Informationsteknik

Chalmers Tekniska Högskola / Göteborgs Universitet

412 96 Göteborg

Telefon: 031-772 1000

The Author grants to Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet.

The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

Omslag:

Illustration av dashboard med interaktiv cirkeldiagram

Institutionen för Data- och Informationsteknik

Göteborg 2021

ABSTRACT

The aim of the project is to develop an application that can visualize statistics from an AGV system using interactive graphs and charts.

The project was carried out in collaboration with AGVE which is a company that develops and builds Automated Guided Vehicles and its control systems. Automated Guided Vehicles (AGV) are self-navigating trucks that transport cargo in warehouses and factories, they are navigated using wires or lasers. AGVE currently has a control system that collects statistics on its AGVs and compiles them into a database.

The application was developed in the programming language R, which is made for statistical data processing, visualization of statistics and data analysis. The Shiny package was also used to turn a standard R application into an interactive web application. The Plotly package was used to generate interactive graphs in the application.

During the planning phase the control system's statistics and database were examined, and with advice from AGVE, it emerged that the three most important categories were to be visualized, these are Utilization, Orders and Alerts.

In the application, the categories were each given a page with user input, graphs and diagrams. The user input allowed the filtering of the statistics with parameters such as date, time and AGV. On the pages, interactive stacked bar charts, pie charts, bubble charts and density maps were used to visualize the statistics.

The completed application enables analysis of the AGV system from several different perspectives. By being user-friendly, giving great freedom to user input and using a number of different graphs that visualize the statistics from different perspectives. The application can be further developed with a comparison page, generation of downloadable reports, increased query efficiency and more.

Keywords: Automated Guided Vehicles, AGV, Dashboard, R, Shiny, Plotly, Visualization, Statistics

SAMMANFATTNING

Projektets syfte är att utveckla en applikation som kan visualisera statistik från ett AGV-system med hjälp av interaktiva grafer och diagram.

Arbetet var gjort tillsammans med AGVE som är ett företag som utvecklar och bygger Automated Guided Vehicles samt dess styrsystem. Automated Guided Vehicles (AGV) är självnavigerande truckar som transporterar last i lager och fabriker, de navigeras med hjälp av slingor eller lasrar. AGVE har idag ett styrsystem som samlar statistik på dess AGV:er och sammanställer det i en databas.

Applikationen var utvecklad i programmeringsspråket R, som är gjort för statistisk databehandling, visualisering av statistik och dataanalys. Paketet Shiny användes också för att förvandla en standard R-applikation till en interaktiv webbapplikation. Plotly-paketet användes för att generera interaktiva grafer i applikationen.

Under planeringsfasen undersöktes AGV-systemets databas och med råd från AGVE identifierades de tre viktigaste kategorierna som skulle visualiseras, dessa är Utilization, Orders och Alerts.

I applikationen fick kategorierna var sin sida med användarinmatning, grafer och diagram. Användarinmatningen tillät filtreringen av statistiken med parametrar som datum, tid och AGV. På sidorna användes interaktiva staplade stapeldiagram, cirkeldiagram, bubbeldiagram och densitetskartor för att visualisera statistiken.

Den färdigställda applikationen möjliggör för analys av AGV-systemet ur flera perspektiv. Genom att vara användarvänlig, ge stor frihet på användarinmatning och använda sig av ett flertal grafer som visualiserar statistiken ur olika perspektiv. Applikationen kan vidareutvecklas med en jämförelsesida, generering av nedladdningsbara rapporter, ökad query effektiviteten med mera.

Nyckelord: Automated Guided Vehicles, AGV, Dashboard, R, Shiny, Plotly, Visualisering, Statistik

FÖRORD

Vi vill tacka Thomas Diding, Christian Skeppstedt och Patrick Malocco från AGVE för att gett oss möjligheten att göra exjobb hos dem. Särskilt Thomas Diding för att ha varit med oss hela vägen och rådgett oss. Vi vill också tacka Ulf Norell, vår handledare från Chalmers som har hjälpt oss under arbetet och klargjort vissa missuppfattningar.

Innehållsförteckning

1. Inledning	1
1.1 Syfte	1
1.2 Mål	1
1.3 Avgränsningar	2
2. Teknisk bakgrund	3
2.1. AGV	3
2.2. BestAGVM	3
2.3. Statistikdatabasen	3
2.3.1. AGV Type	4
2.3.2. Node	4
2.3.3. Orders	4
2.3.4. Alerts	4
2.3.5. Utilization	5
2.4. R	5
2.5. RStudio	5
2.6. R Packages	5
2.7. Shiny	5
2.8. Plotly	5
3. Metod	7
3.1. Programmeringsmiljö	7
3.2. Projektplanering	7
3.2.1 Riktlinjer för User Stories	7
3.3 Applikationsutveckling	8
3.3.1 Undersökning av databasen och dess tabeller	8
3.3.2 Paket	8
3.3.3 Modules	10
3.3.4 Grafer	11
4. Systemkonstruktion	13
4.1 Konstruktion	13
4.2 Databaskommunikation	14
4.3 Modules	14
4.4 Observers och reaktiva värden	14
4.5 Grafplottnig	15
4.6 Global	15
4.7 Deployment	15
5. Resultat	16
5.1 Navigering	16

5.2 Dashboard-sidan	17
5.3 Användarinmatning	22
5.4 Utilization-sidan	24
5.5 Orders-sidan	28
5.6 Alerts-sidan	34
5.7 Settings-sidan	41
5.8 User Stories	44
6. Diskussion/Slutsats	46
6.1 Sammanfattning	46
6.2 Användning	46
6.3 Problem	46
6.4 Etik	47
6.5 Förbättringar och vidareutveckling	47
Källförteckning	49

1. Inledning

AGVE är ett internationellt företag som har kontor och verkstad i Frölunda [1]. Företaget utvecklar, bygger och distribuerar Automated Guided Vehicles (AGV) med tillhörande styrsystem. AGV:s är portabla robotar som förflyttar sig och navigeras med hjälp av till exempel slingor på marken eller lasrar. AGV:s används för transporter av tunga material i industriella sammanhang i fabriker och lager [2].

Företaget är primärt inriktade på att bygga AGV:s till lagerbranschen. De bygger både ett antal standardmodeller och specialbeställda varianter med lastkapacitet som varierar mellan 3 kg upp till 65 ton.

Ett AGV-system genererar stora mängder data i drift, bland annat orderhistorik, felmeddelanden och allmänna statusrapporter. Denna data kan innehålla AGV-id, ordernummer, Node-id, timestamps, hastighet, koordinater, fel-id, med mera. Datan sparas i en sql-databas på kundens server.

1.1 Syfte

AGVE har idag inget sätt att summera och visualisera statistiken som genereras i deras AGV-system. Syftet med projektet är att ta fram diagram och heatmaps som ska underlätta analys och felsökning av systemet.

1.2 Mål

Målet med projektet är att utveckla en applikation som körs lokalt eller på webben med ett grafiskt interface som tillåter användaren att få en insikt och undersöka datan från ett AGV-system. Applikationen ska innehålla flera olika sidor som innehåller olika typer av data. Den ska använda sig av användarinmatning för att generera olika typer av grafer, tabeller och heatmaps beroende på vad användaren har angett.

User stories som ska uppnås med projektet:

User story 1:

Hos ett företag har en alert på en AGV inträffat dagen innan, detta gjorde att AGV:en inte kunde arbeta. En chef ska nu ta reda på vilken alert som inträffat, på vilken AGV, när och hur det har påverkat antalet ordrar avslutade för den AGV:en och AGV-Typen.

User story 2:

En chef på ett företag vet att julen närmar sig och att antalet sålda produkter kommer att öka. Chefen vill ta reda på hur deras AGV-system har påverkats under samma period de föregående åren och hur trenden ser ut.

User story 3:

En chef på ett företag har fått höra från arbetarna på lagret att en av AGV:erna har strulat extra mycket vid en av arbetsstationerna. De vill nu gå och undersöka om detta är sant och i så fall vad problemet kan vara och var det händer.

1.3 Avgränsningar

Projektet kommer inte omfatta arbete med följande punkter:

- Visualisering för all data som är på databasen, utan bara utvald data som är intressant för statistik och analys.
- Utveckling av en webbapplikation med inloggning och admin.
- Någon slags av hårdvara, programvaran kommer vara utvecklad för att köras på Windows-servrar med Microsoft sql-server.
- Deployment och hosting av webbapplikationen,
- Installation eller modifiering av databasen, endast informationshämtning från en redan idriftsatt databas.
- Bildmanipulation för att få heatmaps att mappa med koordinatsystemet.
- HTML och CSS.

2. Teknisk bakgrund

2.1. AGV

Automated guided vehicle, eller AGV, är en självnavigerande truck som används för att förflytta last i lager eller fabrik. AGV:s kan använda sig av olika tekniker för självnavigering. De mest vanligaste är slingor på marken, lasrar, radio och kameror. AGV:s kommer i många olika storlekar och typer, sådana som har en lastkapacitet av 3 kg eller upp till 65 ton, som används i lager, fabriker, sjukhus eller specialarbeten. Se bild nedan.



Fig 1, Bild på en AGV Från [35]. Återgiven med tillstånd

2.2. BestAGVM

BestAGVM är AGVE:s styrsystem som styr alla deras AGV:s hos en kund. Styrsystemet delegerar ordrar till AGV:erna och bevakar deras status. Den sparar också all statistik om AGV:erna på en SQL-databas.

2.3. Statistikdatabasen

Statistikdatabasen är där all data från styrsystemet BestAGVM samlas och sammanställs. De viktiga tabellerna för arbetet är:

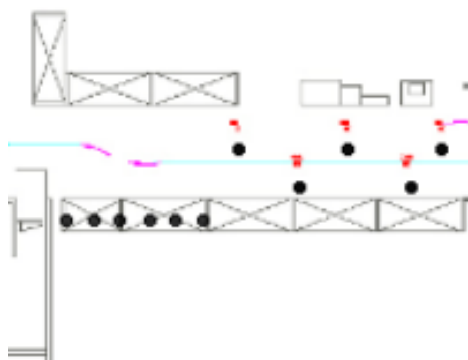
- AGV - Alla AGV:s där deras Id:n, namn och AGV-Type är radade.
- Alert - Alert-information med AGV-id, nod, varaktighet, start- och sluttid och namn.
- Utilization - Utilization-information med AGV-id, nod, varaktighet, start- och sluttid och arbetsläge.
- Order - Order-information med AGV-id, nod, varaktighet, start- och sluttid och ordertyp.

2.3.1. AGV Type

AGV:er som utför liknande arbetsuppgifter brukar samlas ihop till AGV-grupper som kallas AGV-Types. En AGV-Type kan innehålla en AGV som utför en arbetsuppgift ensam, eller upp till ett flertal AGV:er som arbetar med samma sak.

2.3.2. Node

En Node eller nod är en utvald plats i anläggnings-layouten där en AGV kör till eller utför en arbetsuppgift. Noderna hjälper AGV:erna navigera i layouten och arbeta på olika arbetsplatser. En anläggning med ett AGV-system kan ha hundratals till tusentals noder i sin layout. Exempel på Noder finns i figuren nedan.



*Fig 2, Noder i en layout
Varje svart punkt är en Nod i layouten.*

2.3.3. Orders

Ordar är arbetsuppgifter som systemet delegerar till AGV:erna. En order blir avklarad genom att en AGV åker till den specificerade arbetsnoden och utför arbetsuppgiften. Ett exempel är en transportorder där AGV:en åker till en nod för att plocka upp en last och sedan åker till en annan nod för lämna av lasten. I databasen finns sex olika ordertyper: CHARGE, DROP, FIND, GOTO, PARK och TRANSPORT. Varje order har en utgångspunkt i form av en nod, en tidsstämpel för när ordern togs emot samt en kolumn för tiden det tog att genomföra ordern.

2.3.4. Alerts

Alerts är problem som uppstår för AGV:erna under drift, ofta leder det till att de måste pausa sin arbetsuppgift. En alert kan vara så kritisk att en AGV måste gå till reparation, till exempel med en "Current limit exceeded, steer chopper"-alert, eller så ofarlig som en "Front Laser Bumper" som triggas när någon eller något går framför AGV:ens lasersensor. Förutom en beskrivning om vad som inträffat innehåller en alert också information om hur länge problemet varade samt var någonstans problemet inträffade.

2.3.5. Utilization

Utilization är utnyttjandegraden av AGV:erna. Eftersom AGV:erna kan vara i olika arbetslägen samlas då statistik över dessa. Arbetslägena kan till exempel vara: Working, Charging, Manual, Error och Idle. En AGV befinner sig alltid i ett och endast ett arbetsläge. I databasen finns det två olika versioner av tabellen där utnyttjandegraden är grupperad i tidsluckor. En med en upplösning på ett dygn och en med en upplösning på en timme.

2.4. R

R är ett programmeringsspråk använt för statistisk databehandling, visualisering av statistik och dataanalys [3]. R är ett GNU-projekt och är open source.

2.5. RStudio

RStudio är en IDE för R, och används i detta projektet [4]. RStudio har två olika versioner en open source version under AGPLv3-licensen och en kommersiell version. RStudio är utvecklat och underhållet av RStudio PBC som också utvecklar många populära R-paket som ggplot2, dplyr, shiny, RMarkdown med flera.

2.6. R Packages

Inom R-miljön finns det många paket/bibliotek som används för att bygga ut funktionaliteten av R-språket. R har en stor mängd paket för en mångfald olika funktioner, Nanxstats lista [5] är ett exempel på mängden paket inom R och Shiny-miljön.

R har en "centralised software repository" som kallas CRAN (Comprehensive R Archive Network) som används för att tillgängliggöra och ladda ner paket [6].

2.7. Shiny

Shiny är ett R-paket som tillåter utvecklandet av interaktiva webbapplikationer i R. Shiny är utvecklat och underhållet av RStudio PBC [7]. Eftersom det normalt sätt bara är möjligt att bygga R-applikationer som körs lokalt, kan Shiny användas för att få applikationer körandes online i en webb-miljö.

2.8. Plotly

Plotly är ett R-paket som används för att skapa interaktiva grafer i R [8]. Dessa grafer tillåter användaren att sväva och klicka muspekaren på grafer för att få mer information se figur 2 nedan, samt exkludera och urskilja specifika spår. Detta gör graferna mer användarvänliga och mer informativa. Plotly har också ett stort urval av olika graftyper som kan användas i R.

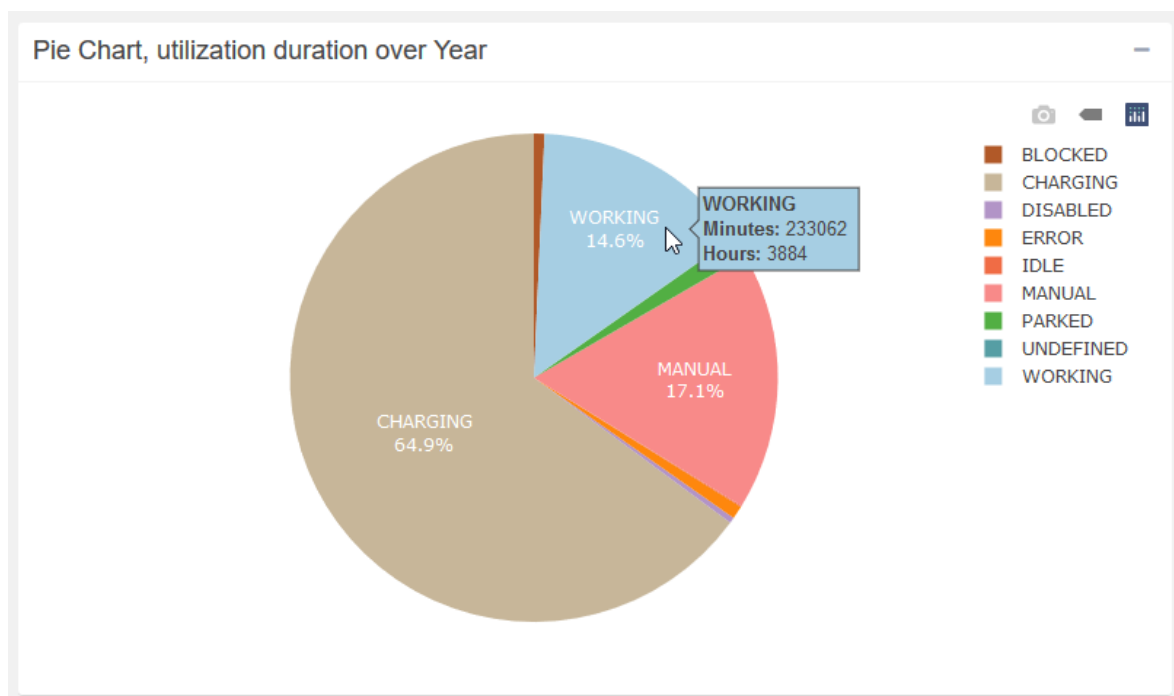


Fig 3, Interaktiv Plotly graf som visar mus svävning.

3. Metod

3.1. Programmeringsmiljö

Projektet genomfördes i programmeringsspråket R med hjälp av IDE:n R Studio. Andra verktyg som använts är bland annat Microsoft SQL Server Management Studio för att underlätta arbete med databasen och Git/Github för versionshantering.

3.2. Projektplanering

Projektarbetet delades upp i följande deluppgifter:

- Lärandet av programmeringsspråket R och dess syntax.
- Undersökning av databasen och dess tabeller. En mer detaljerad uppfattning om systemet skapades och beslut fattades om vilka av tabellerna som är relevanta och kan användas för att rita meningsfulla grafer och heatmaps som kan ge användaren tillräckligt med underlag för att analysera systemet.
- Utredning av olika verktyg och paket inom R som finns tillgängliga och hur de kan användas för att uppnå målet. Fördelar och nackdelar evaluerades innan ett beslut fattades om att arbeta vidare med tekniken eller ej.
- Utveckling av själva applikationen. Utifrån den data och de verktyg som identifierats i föregående steg implementerades olika typer av grafer och inmatningsfält som låter användaren själv ställa in parametrar som exempelvis tidsintervall.

I sektionen *3.3 Applikationsutveckling* nedan beskrivs de här delarna ytterligare.

3.2.1 Riktlinjer för User Stories

För att uppnå de user stories uppsatta i projektmålen följdes ett antal riktlinjer under projektets gång. Dessa var:

- Applikationen ska vara självständig och inte behöva extern mjukvara för att användas.
- Användarinmatningen ska ge stor frihet över filtrering och urval av statistiken. Exempel på parametrar är datum, tid, AGV:er, datatyp samt varaktighet av datatyp. Som beskrivet i sektion *5.8 User Stories* där user story 2 använder sig av datums-inmatning för att undersöka statistik över olika år.
- Plotta statistiken över många olika datakategorier så att användaren kan se statistiken från många olika perspektiv. Exempelvis över tid med veckovis, dygnsvis och timvis, över AGV:er och AGV typer, över noder och på anläggningens layout. Som beskrivet i sektion *5.8 User Stories* där user story 3 använder sig av grafer över anläggningens layout.

3.3 Applikationsutveckling

3.3.1 Undersökning av databasen och dess tabeller

I samråd med AGVE identifierades de tre mest väsentliga tabellerna i databasen: Alerts, Orders och Utilization.

Datan i Alerts är händelsebaserad i den mening att varje alert är en enskild händelse i systemet med en start- och sluttid. Det finns därför två olika sätt att sammanställa alerts av en viss typ, antingen genom att räkna antalet alerts eller genom att summera varaktigheten för varje alert. Båda sätt att räkna ansågs viktiga att presentera eftersom det möjliggör jämförelse mellan de alerts som inträffar mest och de alerts som upptar mest tid. En alert som sker ofta men med en kort varaktighet kan ha en annan påverkan på systemet än en alert som sker mer sällan men har en högre varaktighet. Precis som i Alerts är varje order i Orders en egen händelse och kan därför summeras på samma sätt.

Till skillnad från Alerts och Orders består datan i Utilization inte av enskilda händelser, istället är Utilization uppdelad i två tabeller med olika tidsintervall. Den ena tabellen är strukturerad på så sätt att en rad representerar ett dygn och den totala tiden en AGV befann sig i ett visst arbetsläge under dygnet. Den andra tabellen innehåller samma typ av data men har istället en upplösning på en timme istället för ett dygn. Eftersom en av målsättningarna var att låta användaren justera statistiken på bland annat timintervall valde vi att arbeta med den mer högupplösta tabellen trots att det innebär att många fler rader måste processas för varje query till databasen.

3.3.2 Paket

Eftersom R har en stor mängd paket med många olika funktioner är det väsentligt att man gör ett bra arbete med att undersöka de olika paketen och väljer ut de som passar bäst. För många paket kan leda till en onödigt invecklad och ofokuserad applikation, både för utvecklaren och användaren.

Nedan följer de paket som används inom applikationen.

Shiny

Shiny är det mest centrala paketet i applikationen och valdes eftersom det möjliggör interaktiva grafer i en användarvänlig och lättillgänglig webb-miljö. Fördelen med att designa projektet kring Shiny gentemot andra webb-ramverk är att all utveckling sker inom R-språket genom att abstrahera bort den HTML- och Javascript-kod som annars hade krävts. På så sätt hålls komplexiteten nere och framtida underhåll av applikationen förenklas.

Shinydashboard

Shinydashboard är ett paket som underlättar utveckling av dashboards vilket utgör layouten för applikationen [9].

ShinydashboardPlus

ShinydashboardPlus är en utökning av Shinydashboard med funktioner som exempelvis footers [10].

Fresh

Fresh möjliggör anpassning av applikationen i form av konsekventa färgteman genom hela användargränssnittet [11].

ODBC

ODBC-paketet agerar som ett gränssnitt mellan applikationen och en ODBC-kompatibel databas-driver, i det här fallet Microsoft SQL Server [12]. En anslutning till databasen skapas när applikationen startas och data kan sedan hämtas genom SQL queries.

Dplyr

Dplyr är ett paket som underlättar manipulering och filtrering av data [13]. I applikationen används Dplyrs select-metod för att filtrera bort oönskad data.

ggplot2

Ggplot2 är ett omfattande paket för visualisering och grafitning [14]. Nackdelen med ggplot2 är att graferna som genereras är i form av statiska bilder som därför inte är interaktiva. För majoriteten av grafer i applikationen används därför paketet Plotly för datavisualisering.

Plotly

Plotly är, precis som ggplot2, ett paket för datavisualisering men tillåter också interaktion [15]. Alla grafer i applikationen med undantag för bubbeldiagrammen och densitetskartorna är genererade med hjälp av Plotly.

Shinyjs

Shinyjs expanderar Shinys UI-element med fler funktioner [16]. I applikationen används detta för att förbättra användarvänligheten genom att bland annat dynamiskt gömma eller inaktivera inmatningsfält beroende på kontext.

RColorBrewer

RcolorBrewer är ett paket som används för generering av färgpaletter [17]. Inom applikationen används paletterna för att färgkoda olika datakategorier som exempelvis ordertyper. Viktiga egenskaper hos en palett är att färgerna som används är distinkta från varandra och att data som förekommer i flera olika grafer representeras av en och samma färg.

Jpeg

Jpeg-paketet gör det möjligt att läsa och skriva bilder i JPEG-format [18]. I applikationen används readJPEG-metoden för att läsa in layout-ritningen som heatmaps sedan kan plottas på.

ShinyBS

ShinyBS lägger till komponenter och funktionalitet ifrån Bootstrap i Shiny [19]. En komponent som används ifrån ShinyBS är Tooltip-komponenten.

Shinycssloaders

Shinycssloaders gör applikationen mer användarvänlig genom att visa en spinner-animation på webbsidan medan statistik hämtas och en graf genereras eller uppdateras [20]. På så sätt får användaren en bättre indikation på att ny information håller på att hämtas. Se videon i början av 5. Resultat för en demonstration av spinners i applikationen.

DT

DT (DataTables) möjliggör generering av datatabeller i HTML-format i R vilket används för att presentera AGV-tabellen på dashboard-sidan i applikationen [21].

Lubridate

Lubridate underlättar hanteringen av datum och tid i R genom en uppsättning av parsning och beräknings-metoder [22].

3.3.3 Modules

Någorlunda tidigt i utvecklingsstadiet märktes att i en standard Shiny-applikation var det omöjligt att använda en plot eller komponent flera gånger i samma applikation. Detta skapade problem eftersom vissa komponenter som skulle kunna användas flera gånger istället fick utvecklas genom kodduplikation.

Problemet löstes efter upptäckten av Modules vilket är en funktionalitet i Shiny [23]. Modules möjliggör för applikationer att använda en eller flera nya namespaces istället för det globala namespace. Detta tillåter komponenter att ha samma namn i olika moduler och med detta kunna användas flera gånger utan kodduplikation.

Med den nya förståelsen av Modules fick applikationen uppdateras till den nya strukturen. Ett exempel på en komponent som används flera gånger är användarinmatnings-boxen som används flera gånger på de olika statistiksidorna.

3.3.4 Grafer

För att visualisera datan hämtad från databasen behövdes ett val av graftyper göras.

Staplat stapeldiagram eller Stacked bar chart på engelska är en graftyp som expanderar funktionaliteten på ett standard stapeldiagram genom att stapla flera staplar på varandra. De används för att jämföra numeriska värden mellan subkategorier och kategorier. Se figuren nedan.

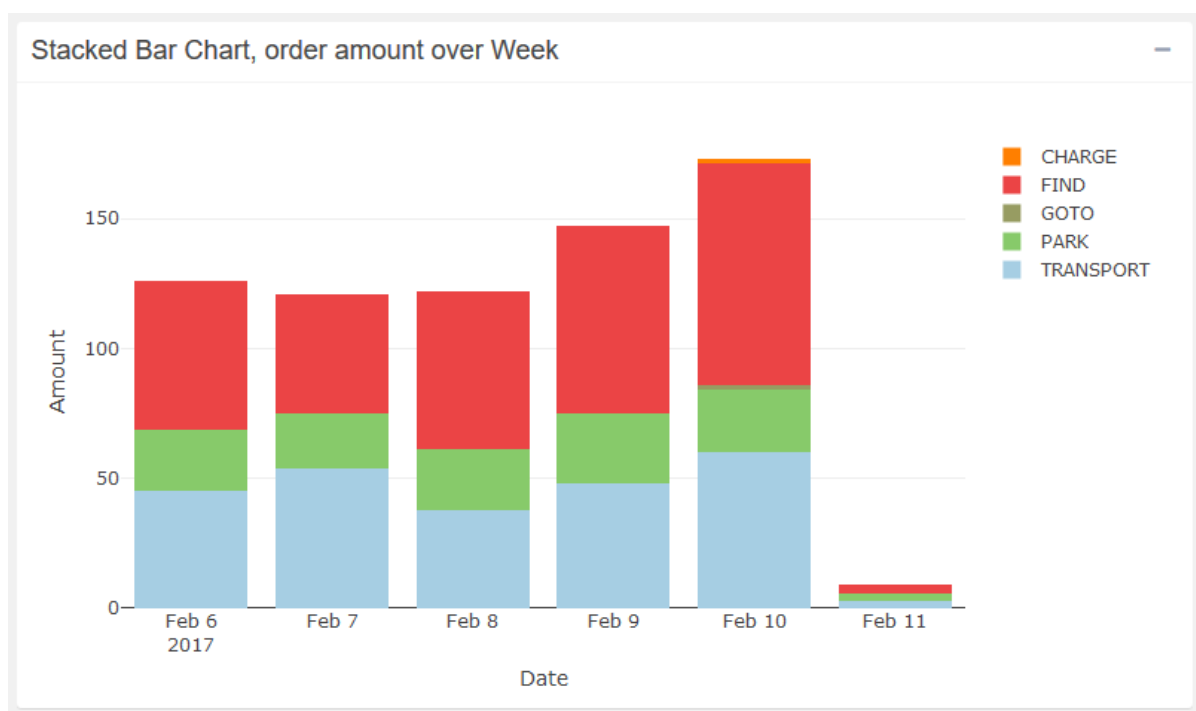


Fig 4, Exempel på en staplat stapeldiagram.

Det märktes under undersökningen av olika graftyper att staplat stapeldiagram skulle passa bra på flera ställen i applikationen där statistiken skulle plottas. Detta eftersom många av graferna i applikationen behöver en axel för tid, en axel för mängd och en dimension för kategorier.

Heatmap är en graftyp för att plotta numerisk magnitud på två dimensioner med färg. En av subtyperna för heatmaps är densitetskarta eller density map på engelska, som mappar densiteten av punkter på en karta med färg. Det är dessa densitetskartor som används i applikationen för att mappa mängden Alert och Order-händelser på en karta. Se figuren nedan.



Fig 5, Exempel på en densitetskarta med en layout som bakgrundsbild.

Bubbeldiagram eller Bubble chart på engelska är en graftyp för att mappa tredimensionell data genom x och y axeln samt storleken på bubblorna. Graftypen är en utveckling på en scatterplot med en extra dimension.

Det är bubbeldiagram som används i applikationen för att mappa mängden tid per alert/order på en karta där en större bubbla representerar en längre tidsåtgång. Se figuren nedan.

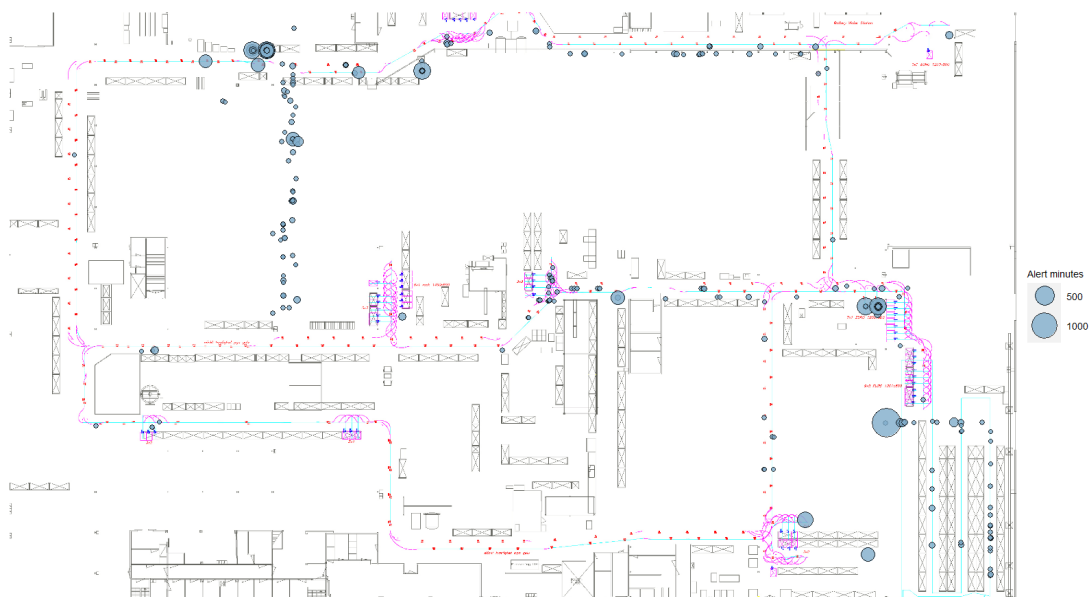


Fig 6, Exempel på ett bubbeldiagram med en layout som bakgrundsbild.

4. Systemkonstruktion

4.1 Konstruktion

En applikation som är byggd med Shiny är uppdelad till en UI-del, en Server-del och en Global-del.

UI-delen hanterar layout, teman, navigering, statiska komponenter och avgör var på sidan de dynamiska komponenterna ska skapas.

Server-delen hanterar serverspecifika reaktiva värden, observers, grafplottning och skapandet av dynamiska komponenter. Den gör detta med ett antal render-funktioner som till exempel `renderPlot`, `renderText` eller `renderUI`.

Global-delen hanterar globala variabler, globala reaktiva värden och initialisering.

Förutom detta har också UI-delen och Server-delen delats upp till flera filer för olika fokusområden. Till exempel `alertUI` för Alerts-sidan med en korresponderande `alertServer` med server-funktionerna för den sidan. De andra sidorna är uppdelade på samma sätt med filer som `orderUI` och `orderServer`, `settingsUI` och `settingsServer` och så vidare. Se figuren nedan för hela filstrukturen.

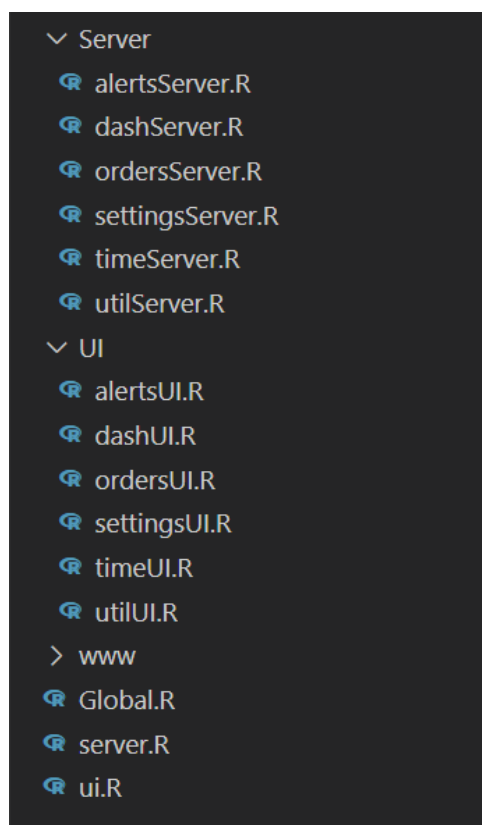


Fig 7, Filstrukturen för applikationen.

4.2 Databaskommunikation

Applikationen kommunicerar med statistikdatabasen genom funktioner i R-paketet odbc [12]. Vid applikationens start skapas en förbindelse med databasen genom att ange namn, driver och platsen för databasen. Genom denna förbindelse kan R-applikationen kommunicera med databasen.

För databas-queries som har användarinmatning används parametriserade queries vilket skyddar mot injektionsattacker [24].

4.3 Modules

Modul-funktionaliteten har implementerats på alla ställen i koden där en komponent används flera gånger. Ett exempel är inmatnings-boxen som används fyra gånger per statistiksida, alltså en gång per tab i dess box och totalt 12 gånger i hela applikationen. Varenda en av statistiksidorna har också en modul per tab av inmatnings-boxen. Detta är för att graferna återanvänds för år, månad, vecka och dag.

4.4 Observers och reaktiva värden

Shiny är uppbyggd för att använda reaktivitet med sin kommunikation mellan UI och Server [25]. Detta betyder att när en användare använder en inmatnings-komponent skapas ett event som triggar en observer som lyssnar på det eventet.

I applikationen används observers för att lyssna på ändringar i datumspann och andra filter för att sedan uppdatera både graferna men också andra inmatningsfält. Ett exempel är när användaren ändrar det valda året. Då lyssnar en observer efter ändringar i år-fältet och uppdaterar sedan alternativen i månads-fältet med de månader det finns data för i systemet och för det nya angivna året.

Reaktiva värden är variabler som triggar events när dess värde ändras vilket propagerar förändringen till alla platser i koden där variabeln refereras.

4.5 Grafplottning

- För cirkeldiagrammen i applikationen används Plotly Pie Charts [26].
- För stapeldiagrammen och de staplade stapeldiagrammen används Plotly Bar Charts[27].
- Färgerna för graferna kommer från RcolorBrewer [17] som har en interaktiv palettväljare i dess dokumentation vilket varit hjälpsamt för val av färger[28].
- För genereringen av bubbeldiagrammen används ggplot2 geom_point [29].
- För genereringen av densitetskartorna används ggplot2 geom_bin2d [30].

4.6 Global

Global-delen är där applikationen initieras genom att sätta upp variabler, reaktiva värden, importera paket, skapa förbindelse med databasen och övriga inställningar.

4.7 Deployment

I dagsläget har var och en av AGVE:s kunder ett eget styrsystem och databas som körs på en Windows-server hos kunden. Därför skulle det vara bra om projektets webbserver också kör på samma maskin som resten av mjukvaran så att allt är centralt och att den har lättillgängligt till databasen. För deployment av applikationen finns det tre olika alternativ Shinyapps.io, Shiny Server eller Shiny Proxy.

Shinyapps.io är en cloud baserad hostingplattform för shiny applikationer utvecklad av RStudio teamet [31]. Men för AGVE skulle inte Shinyapps.io fungera bra eftersom de vill helst själva hosta sin egen mjukvara.

Shiny Server är en webbserver för lokalt användning och är också utvecklad av RStudio teamet [32]. För Shiny Server finns det också två olika versioner, en open source version med stora restriktioner för användning, och en betald version som heter RStudio Connect med större funktionalitet. Dock så är båda bara körbara på linux som inte skulle fungera för AGVE om de inte gör virtuella maskiner.

Shiny Proxy är en open source webbserver för shiny som körs serverside på java [33]. Shiny Proxy körs också genom Docker med Docker images, detta gör att webbservern kan köra på windows genom Docker och WSL.

Eftersom projektet inte omfattar deployment eller hosting av applikationen är detta bara förslag, och slutbeslutet och arbetet med deployment får AGVE göra själva.

5. Resultat

Applikationen har fem olika sidor med olika funktioner.

Först Dashboard-sidan som visar en översikt av statistiken för de närliggande dagarna.

De tre statistiksidaorna Alerts, Utilization och Orders som visar statistiken för de utvalda databastabellerna som undersöktes, se sektion 3.3.1 *Undersökning av databasen och dess tabeller*.

Sist Settings-sidan som har inställningarna för applikationen.

Med de fem olika sidor och dess grafer inuti kan en användare analysera och felsöka ett AGV-system. Mer detaljerad information om sidorna kommer i sektionerna nedan och exempel på hur applikationen kan används kommer i video nedan och sektionen 5.8 *User Stories*.

Demonstrationsvideo för hela applikationen med många av dess funktioner och utseende:
<https://www.youtube.com/watch?v=bT2gOGqCDng>

5.1 Navigering

Applikationen använder sig av shinydashboard och shinydashboardplus-paketen för att skapa en dashboard-layout. Denna layout har en sidebar med navigeringen för webbsidan. Applikationen har i nuläget fem olika sidor som markeras blåa när man är på respektive sida. Se figuren nedan.

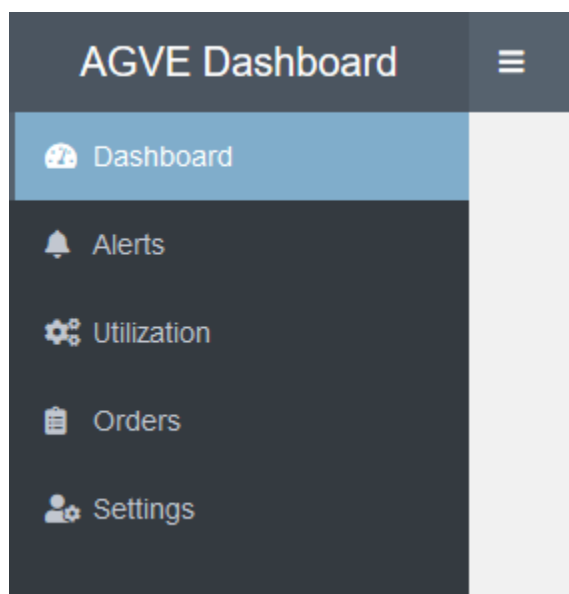


Fig 8, Webbsidans Navigering.

*Visar headern, sidebarn samt de valbara sidorna:
Dashboard, Alerts, Utilization, Orders och Settings med deras ikoner.*

5.2 Dashboard-sidan

Dashboard-sidan är hemsidan för webbapplikationen. Det är hit man kommer när man går in på webbsidan och det är också här man kan få en översiktlig bild av statistiken för de närliggande dagarna. Se figuren nedan för hela Dashboard-sidan.

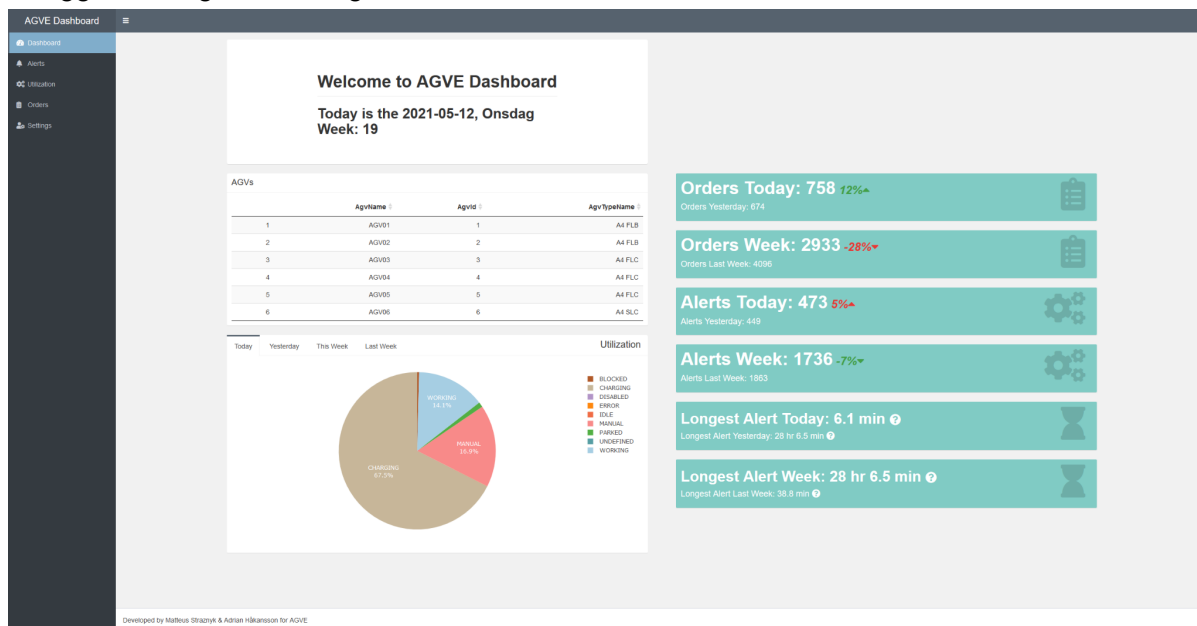


Fig 9, Dashboard-sidan.

Visar uppifrån och ner, vänster till höger:
 Välkommen-boxen, AG-tabellen, Utilization pie chart,
 Order-infoboxarna och Alert-infoboxarna.

Välkommen-boxen mottar användare och informerar om dagens datum, veckodag och veckonummer. Eftersom applikationen ska kunna användas helt fristående utan en klocka eller kalender så finns denna information på webbsidan. Boxen kommer från shinydashboard-paketet och datumen hämtas från systemdatumet och omvandlas sedan med funktioner till veckodag och veckonummer från lubridate-paketet. Se figuren nedan.

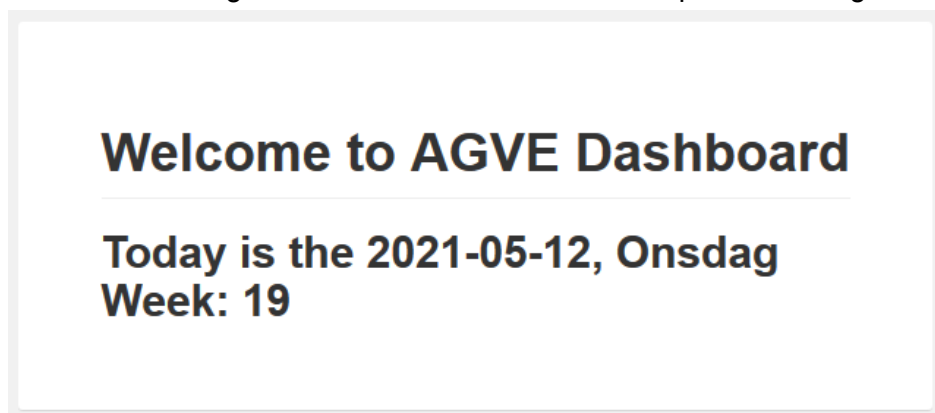


Fig 10, Välkommen boxen.

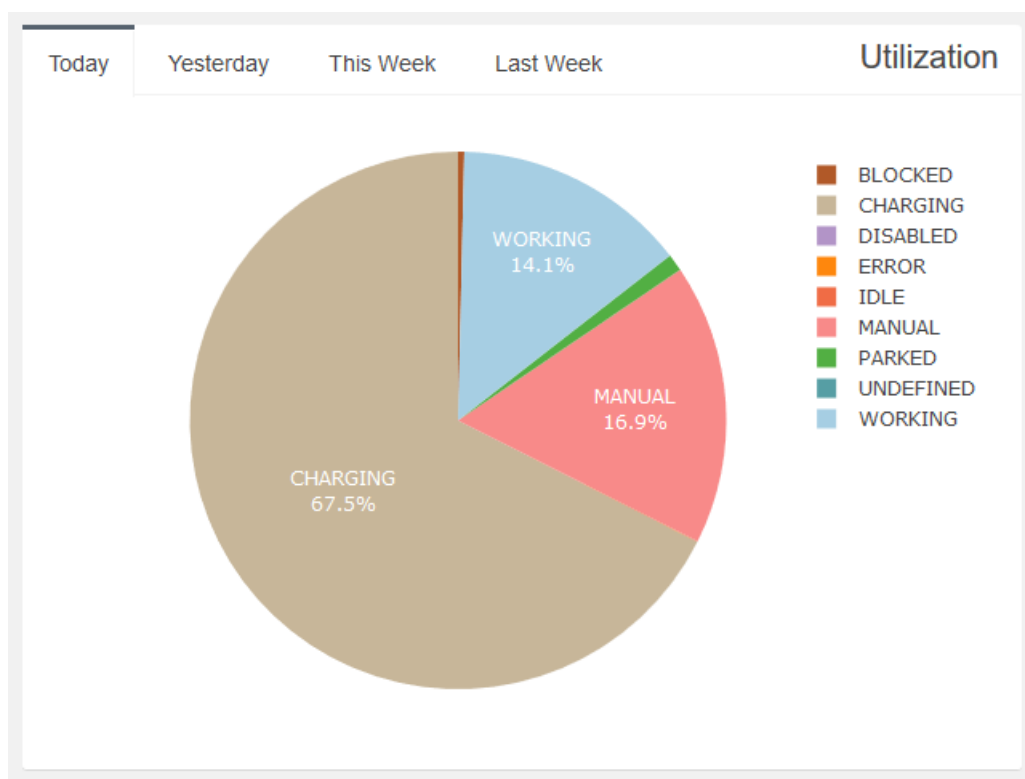
Innehåller en välkomst-text samt datum, veckodag och veckonummer.

AGV-tabellen informerar användaren om alla AGV:erna som finns i systemet genom att hämta informationen från statistikdatabasen. Den visar Id, namn, typ och tillåter sortering efter var och en av kolumnerna. Eftersom applikationen ska kunna vara helt självstände behövs ett sätt att hålla koll på AGV:erna och det är denna tabellen hjälper till med. För skapandet av tabellen användes DT-paketet. Se figuren nedan.

AGVs			
	AgvName ↕	AgvId ↕	AgvTypeName ↕
1	AGV01	1	A4 FLB
2	AGV02	2	A4 FLB
3	AGV03	3	A4 FLC
4	AGV04	4	A4 FLC
5	AGV05	5	A4 FLC
6	AGV06	6	A4 SLC

*Fig 11, AGV tabell.
Tabell av AGV:er med kolumnerna
AGVName, AGVId, AGVTypeName.*

Utilization-cirkeldiagrammet används för att ge användaren en förståelse om utnyttjandegraden av AGV:erna för de följande tidsintervallen: idag, igår, nuvarande vecka och föregående vecka. En tabBox från shinydashboard-paketet används för box-tabbarna. Vardera en av tabbarna har ett korresponderande cirkeldiagram för det tidsspannet. Plotly används för genereringen av cirkeldiagrammet och datan var hämtad från Utilization-tabellen i statistikdatabasen. Se figuren nedan.



*Fig 12, Utilization pie chart.
TabBox med fyra tabbar:
Today, Yesterday, This Week, Last Week,
och en korresponderande cirkeldiagram per tab.*

Order info-boxarna visar för användaren mängden avslutade ordrar för olika tidsspann och den procentuella förändringen jämfört med föregående dag respektive vecka. Info-boxarna var gjorda med shinydashboard och Shiny's renderText-funktion för de dynamiska siffrorna och procenten. Denna information var hämtad från statistikdatabasens Order-tabell. Se figuren nedan.

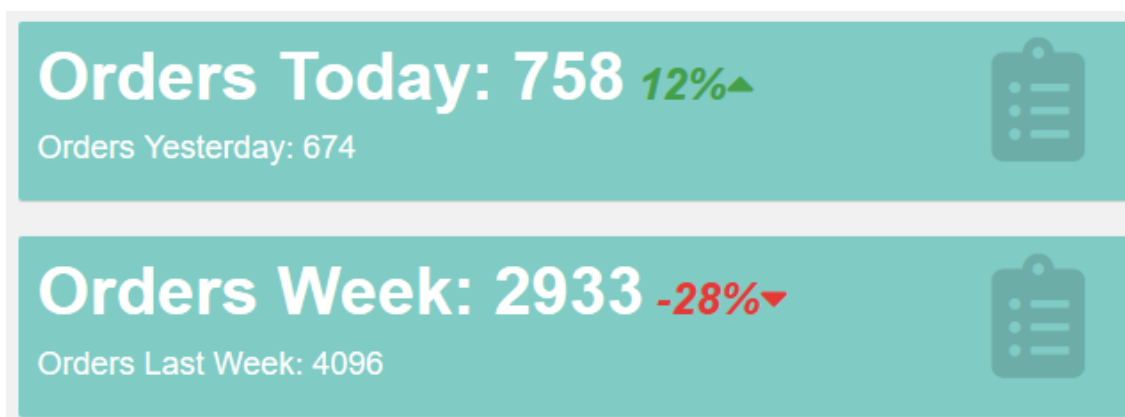


Fig 13, Order info-boxar.

Övre boxen innehåller antalet ordrar för idag, igår och skillnaden i procent.

Nedre boxen innehåller antalet ordrar för nuvarande och föregående vecka och skillnaden i procent.

Alert info-boxarna är fyra boxar som informerar om Alert-statistiken för de närliggande dagarna. De två översta boxarna visar samma information som Order-boxarna fast för alerts. De nedre två visar längden på den längsta alerten under respektive tidsspann. De har också en ikon som visar en tooltip när muspekaren svävar med mer information för just den alerten. All data som visas blir hämtad från Alert-tabellen på statistikdatabasen. Se figuren nedan.

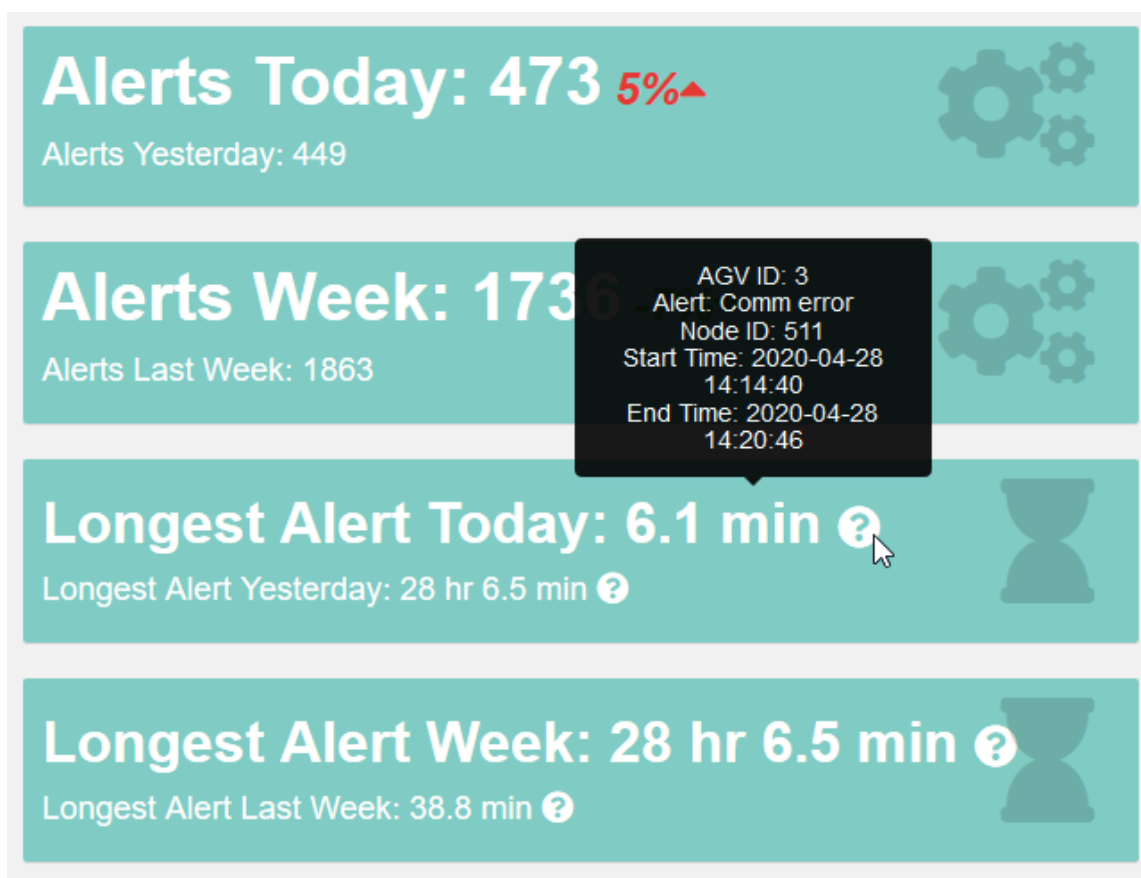


Fig 14, Alert info-boxar.

Övre två boxarna visar antalet alerts för idag, igår, denna och föregående veckan och skillnaden i procent.

De undre två visar längsta alerten under samma tidsspann, med muspekaren på tooltip ikonen som visar mer information om den alerten.

5.3 Användarinmatning

På varje av de tre statistiksidorna finns det en inmatnings-box. Det är i den användaren matar in all information som ska användas för att generera graferna.

The screenshot shows a web interface for utilization data. At the top, there are four tabs: 'Utilization Year', 'Utilization Month', 'Utilization Week', and 'Utilization Day'. The 'Utilization Year' tab is selected. Below the tabs, there is a 'Year' dropdown menu showing '2020'. Underneath is a section titled 'Active Hours' with a horizontal slider ranging from 0 to 24. The slider has major ticks every 3 units and minor ticks every 1 unit. Two circular handles are positioned on the slider at 7 and 20. Below the slider is a checkbox labeled 'Advanced hour selector'. At the bottom is a section titled 'AGV Choice' with three radio button options: 'By All' (selected), 'By Agv Type', and 'By AGV ID'.

Fig 15, Användarinmatnings-box.

Visar uppifrån och ner:

Tids-tabbar, tids-dropdown, aktiva timmar-slidern,
advanced hour selector checkmark, AGV choice.

Överst i boxen finns tabbar som används för att ställa in datumspannet som statistiken ska omfatta. Dessa är årsvis, månadsvis, veckovis och dygnsvis. När en av tabbarna är vald visas den korresponderande dropdownmenyn med dess datum nedanför.

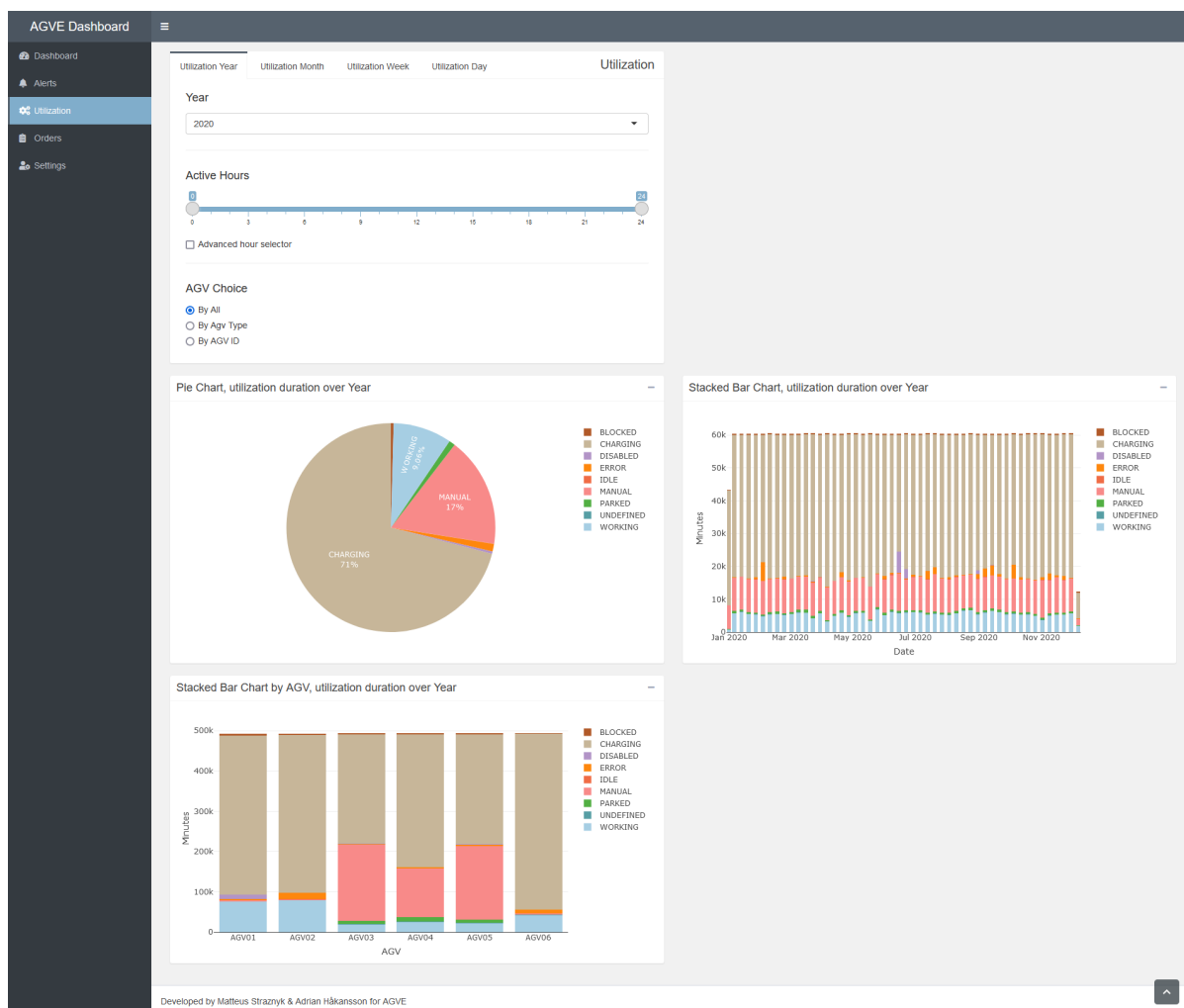
I mitten av boxen ligger "aktiva timmar"-slidern. Den styr vilka timmar under dygnet som statistiken ska komma ifrån och plottas. Det är en 24 timmars-slider med en starttid och sluttid på dygnet. Under slidern finns en "advanced hour selector"-kryssruta som expanderar "aktiva timmar"-slidern till att visa en slider för varje dag i veckan.

Nederst i boxen finns radioknappar för att välja vilka AGV:er som ska vara med i statistikdatan som ska plottas. Valen är alla AGV:er, per AGV-typ eller per AGV-Id. Detta gör det möjligt för användaren att plotta grafer på statistiken för en AGV en AGV-typ eller alla AGV:er.

Det som beskrivits ovan är den standardiserade inmatnings-boxen och finns på alla tre av statistiksidorna. Utöver detta finns det också extra filter på Alert-sidan som tillåter större filtrering av datan, vilket beskrivs i sektion *5.6 Alerts-sidan*.

5.4 Utilization-sidan

Utilization-sidan är sidan där statistiken från Utilization-tabellen på statistikdatabasen visualiseras. Sidan använder sig av användarinmatnings-boxen för full frihet av tid och AGV-filtrering, samt tre olika grafer som visar statistiken över tid, per AGV-Id eller AGV-Type och som en totalsumma i formen av ett cirkeldiagram. Se hela sidan på figuren nedan.



*Fig 16, Utilization-sidan.
Visar vänster till höger, uppifrån och ner:
Användarinmatnings-boxen,
Utilization-cirkeldiagram,
Utilization staplat stapeldiagram över tid,
Utilization staplat stapeldiagram per AGV.*

Utilization-cirkeldiagrammet visar den totala tidsfördelningen mellan arbetslägena i procent för den valda användarinmatningen. Eftersom grafen är gjord med Plotly är den också interaktiv vilket tillåter användaren att sväva muspekaren över pajskivorna för att få mer information över det arbetsläget. Se figuren nedan.

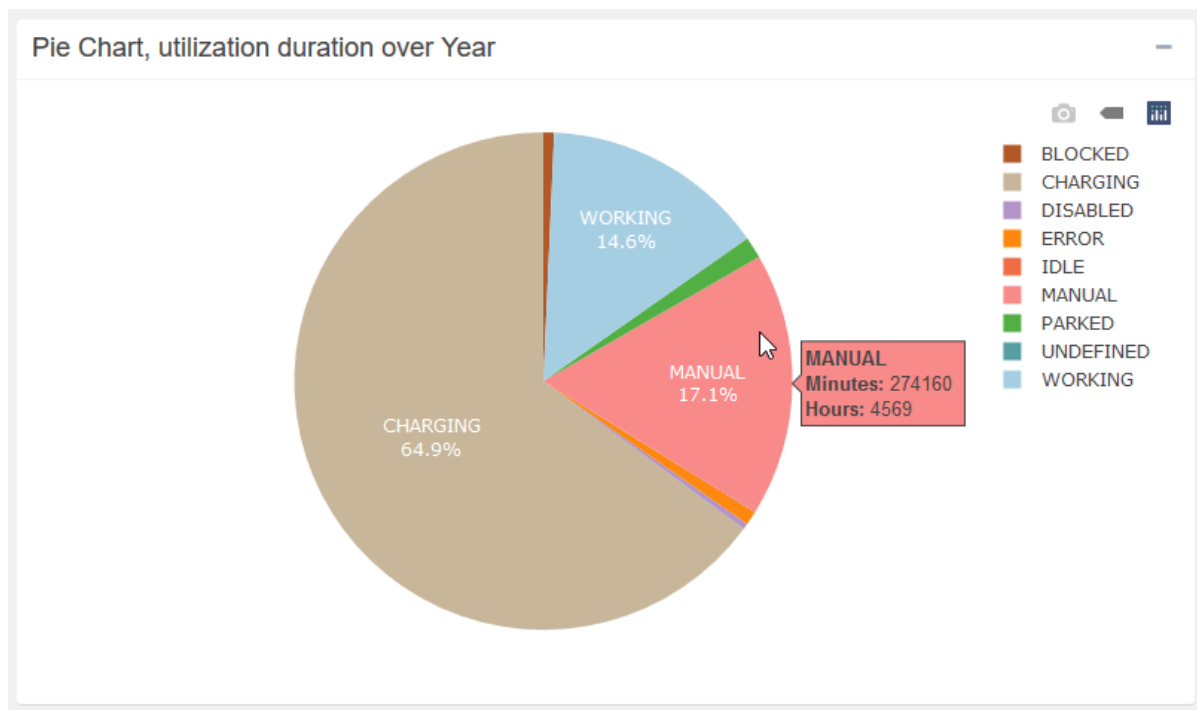


Fig 17, Utilization cirkeldiagram.

Utilization staplat stapeldiagram per AGV visar längden tid de olika AGV:erna spenderade i de olika arbetslägena för den valda användarinmatningen. Varje stapel korresponderar till en AGV och är uppdelad i de olika arbetslägena med färg. Interaktiviteten tillåter användaren av sväva muspekaren över staplarna och ta bort specifika typer från grafen genom att klicka på texten i diagrammets teckenförklaring. Se figuren nedan.

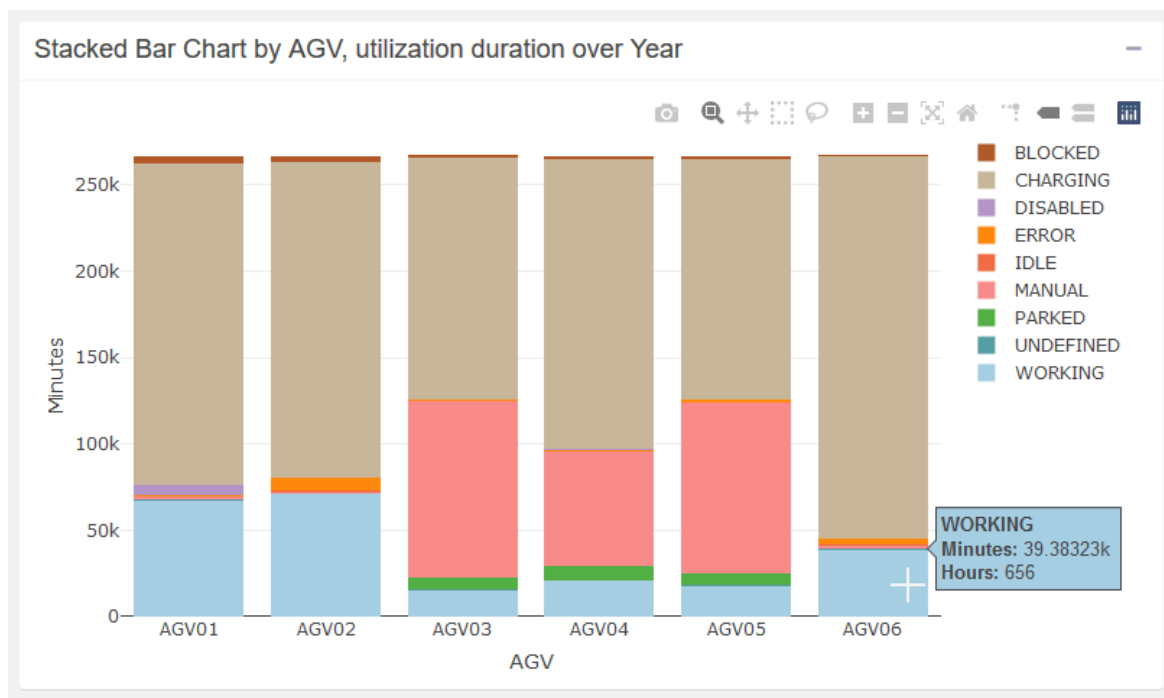
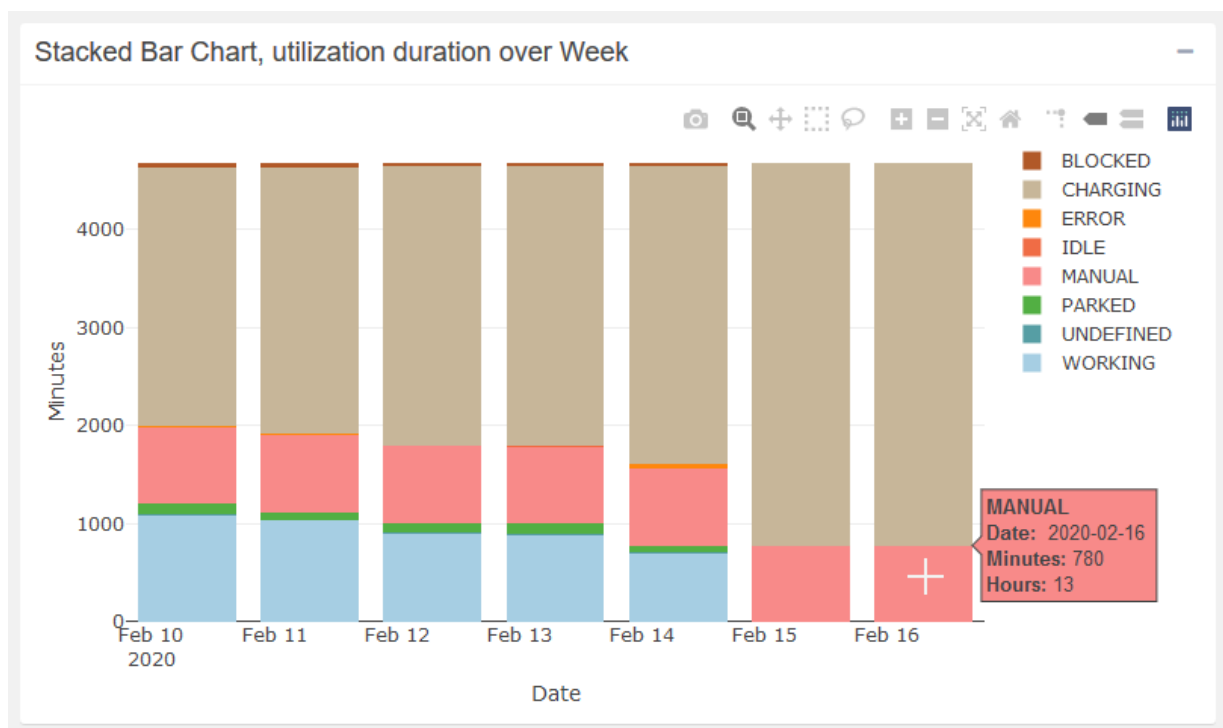


Fig 18, Utilization staplat stapeldiagram per AGV.

Utilization staplat stapeldiagram över tid är lik *Utilization staplat stapeldiagram per AGV* från föregående sida. Men i skillnad är staplarna grupperade på datum i x-axeln, där varenda stapel korresponderar till en vecka, dygn eller timme beroende på vilken användarinmatning som är vald. Detta betyder att grafen visar utnyttjandegraden över tid, som kan användas för att se trender över tid. Se figuren nedan.



*Fig 19, Utilization staplat stapeldiagram över tid.
Där varenda stapel korresponderar för utnyttjandegraden för en dag.*

5.5 Orders-sidan

Orders-sidan är sidan där statistiken från Order tabellen på statistikdatabasen visualiseras. Sidan använder sig av användarinmatnings-boxen för full frihet av tid och AGV-filtrering, samt olika grafer för att visualisera statistiken. Ordrrar visas både som en mängd och varaktighet. Den gör detta på samma sätt som i Utilization där det finns en graf för totalsumman, över tid och per AGV-Id eller AGV-type. Men också ett bubbeldiagram och en densitetskarta som mappar statistiken över anläggningens layout. Se hela sidan på figurerna under.

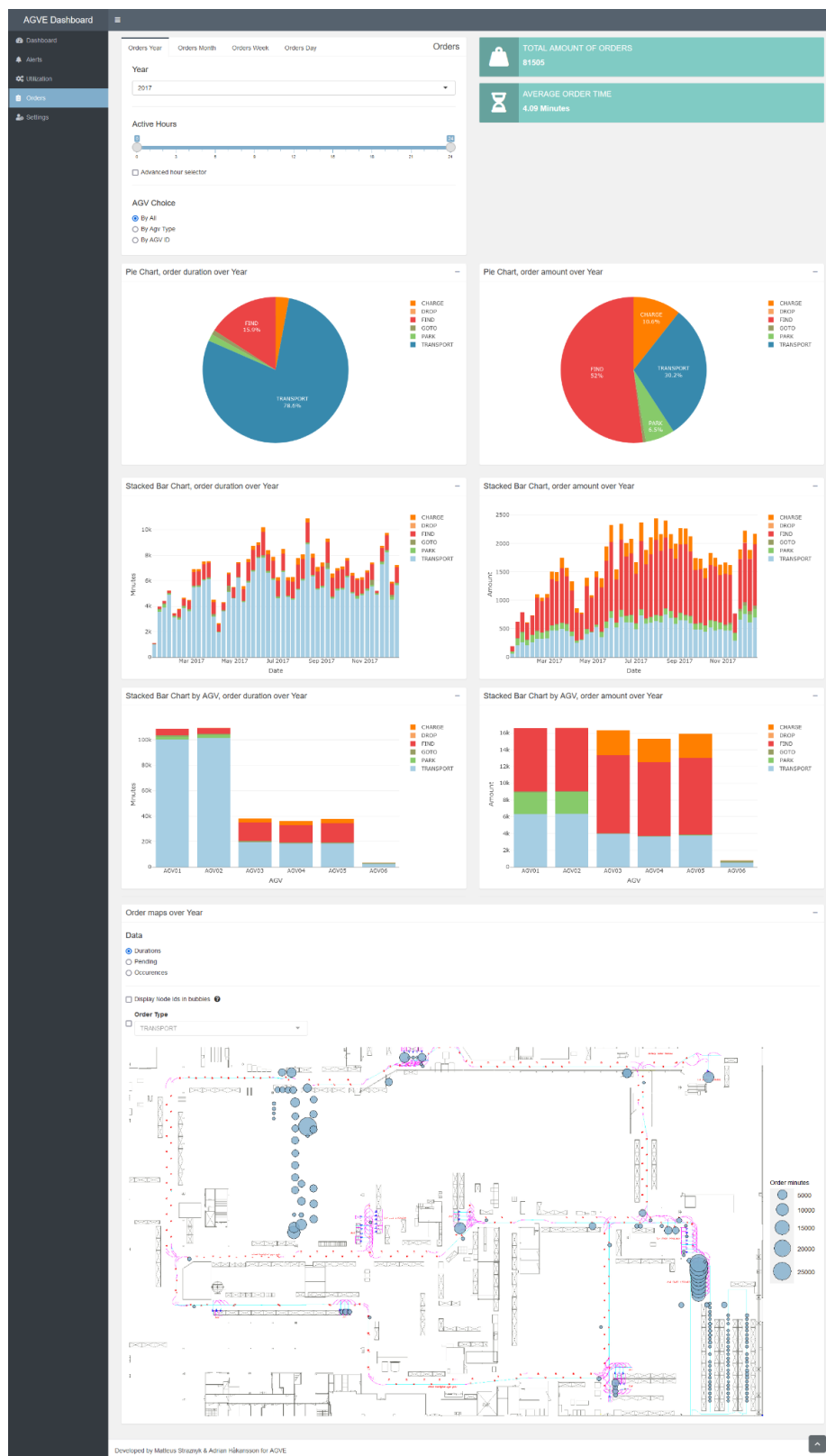
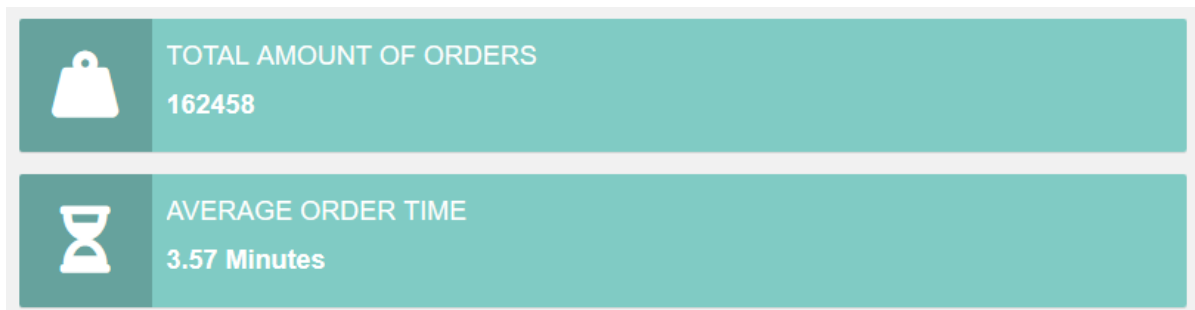


Fig 20, Orders-sidan
 Visar vänster till höger,
 uppifrån och ner:
 Användar inmatnings boxen,
 Orders total amount info box,
 Orders average time info box,
 Orders duration cirkeldiagram,
 Orders amount cirkeldiagram,
 Orders duration staplat
 stapeldiagram över tid,
 Orders amount staplat
 stapeldiagram över tid,
 Orders duration staplat
 stapeldiagram per AGV,
 Orders amount staplat
 stapeldiagram per AGV,
 Orders bubble/density map.

Bredvid användarinmatnings-boxen ligger två infoboxar som kan ge en övergripande bild av situationen utan att behöva iaktta graferna nedan. Det är användbart för en användare som snabbt byter mellan olika inmatningar och med detta inte behöver ta tiden för att betrakta graferna.



*Fig 21, Orders-infoboxar.
Den övre visar totala mängden ordrar,
den undre visar den genomsnittliga order tiden.*

De första graferna på sidan är två cirkeldiagram (*Orders duration* och *Orders amount*) som visar fördelningen mellan ordertyperna under det valda tidsintervallet och de valda AGV:erna. I den vänstra grafen är datan grupperad efter den totala varaktigheten för respektive ordertyp medan datan i den högra grafen är grupperad efter totala antalet ordrar. Genom att jämföra de två graferna kan man snabbt få en uppfattning om hur ofta olika ordrar inträffar och dess genomsnittliga varaktighet.

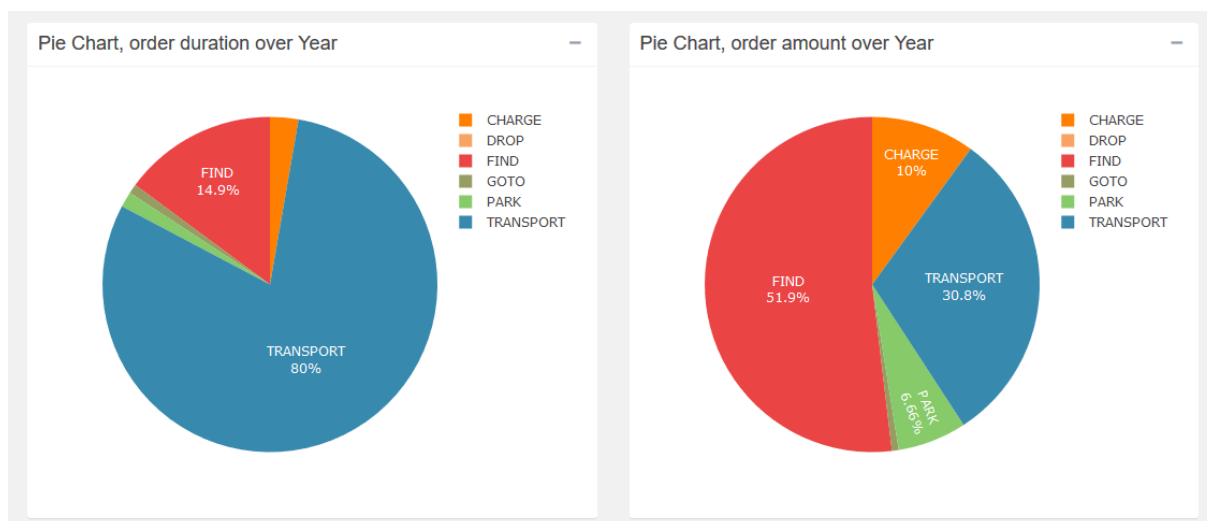


Fig 22, Orders duration och Orders amount cirkeldiagram.

I de staplade stapeldiagrammen visas också ordarnas varaktighet och antal, men uppdelat över tid på x-axeln. I de två graferna nedan visas statistik för ett helt år, varje stapel representerar en vecka och är indelad i de olika ordertyperna enligt färgkodningen.

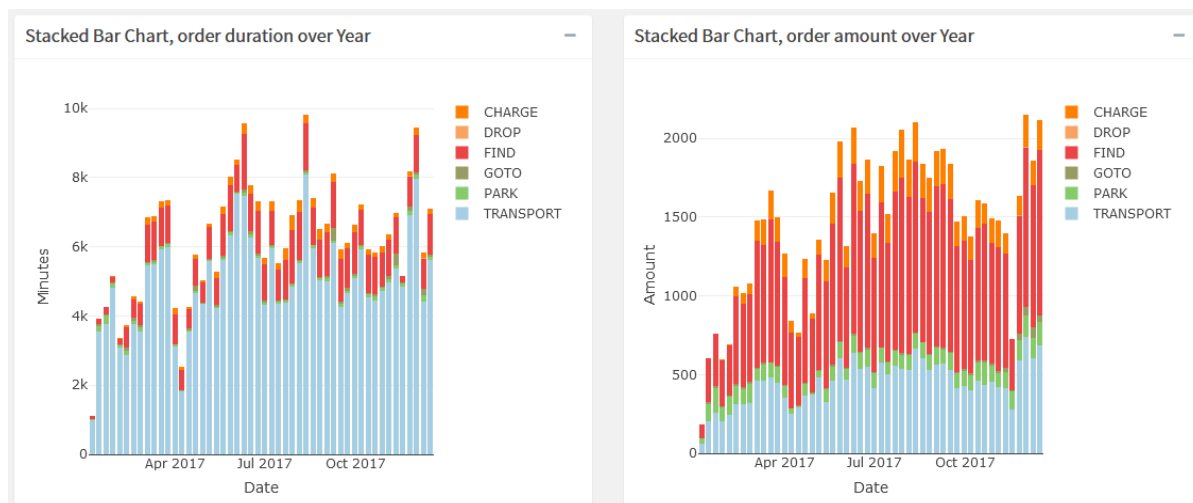


Fig 23, Orders duration över tid och Orders amount över tid staplat stapeldiagram.

De två sista stapeldiagrammen på sidan är uppdelade över AGV:erna på x-axeln istället för tid som i föregående grafer. Här kan man jämföra fördelningen av ordrar mellan AGV:er eller AGV-typer.

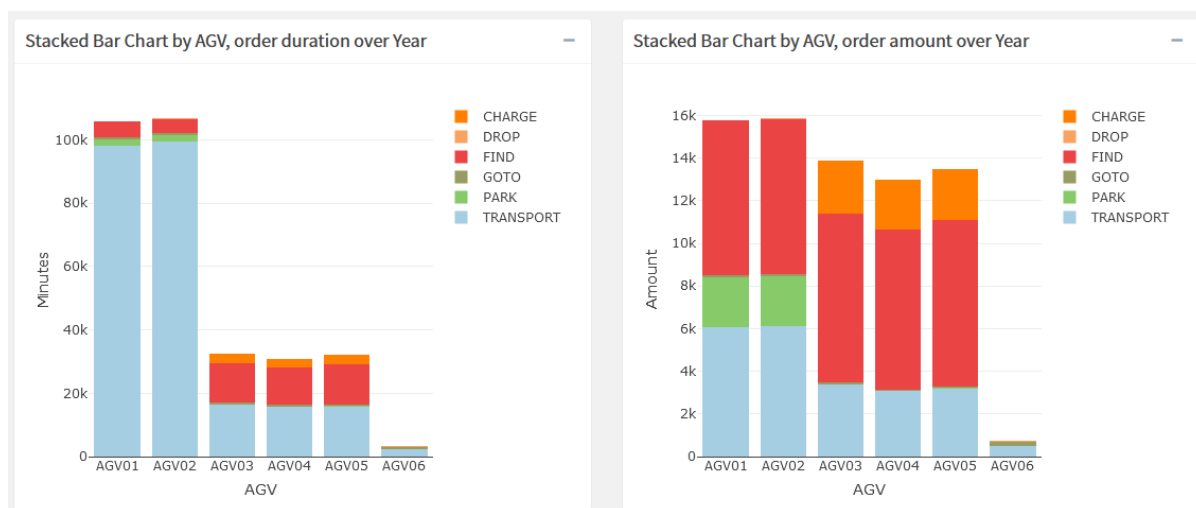


Fig 24, Orders duration per AGV och Orders amount per AGV staplat stapeldiagram.

Heatmaps används för att presentera tre olika typer av data över layoutens koordinatsystem. Dessa är ordarnas varaktighet (Durations), totala antalet ordrar (Occurrences) och dröjtiden från att en AGV tog emot en order till att ordern aktivt påbörjats (Pending). Användaren kan välja att se data från alla ordertyper eller välja enbart en ordertyp genom en dropdown-box.

Durations och Pending presenteras med hjälp av bubbeldiagram där varje bubbla representerar en eller flera grupperade ordrar som har utgångspunkt i samma koordinater och där storleken på bubblan är proportionerlig med varaktigheten. Till höger om layouten finns en teckenförklaring där olika storleksnivåer är mappade till ett värde i minuter. Se figuren nedan.

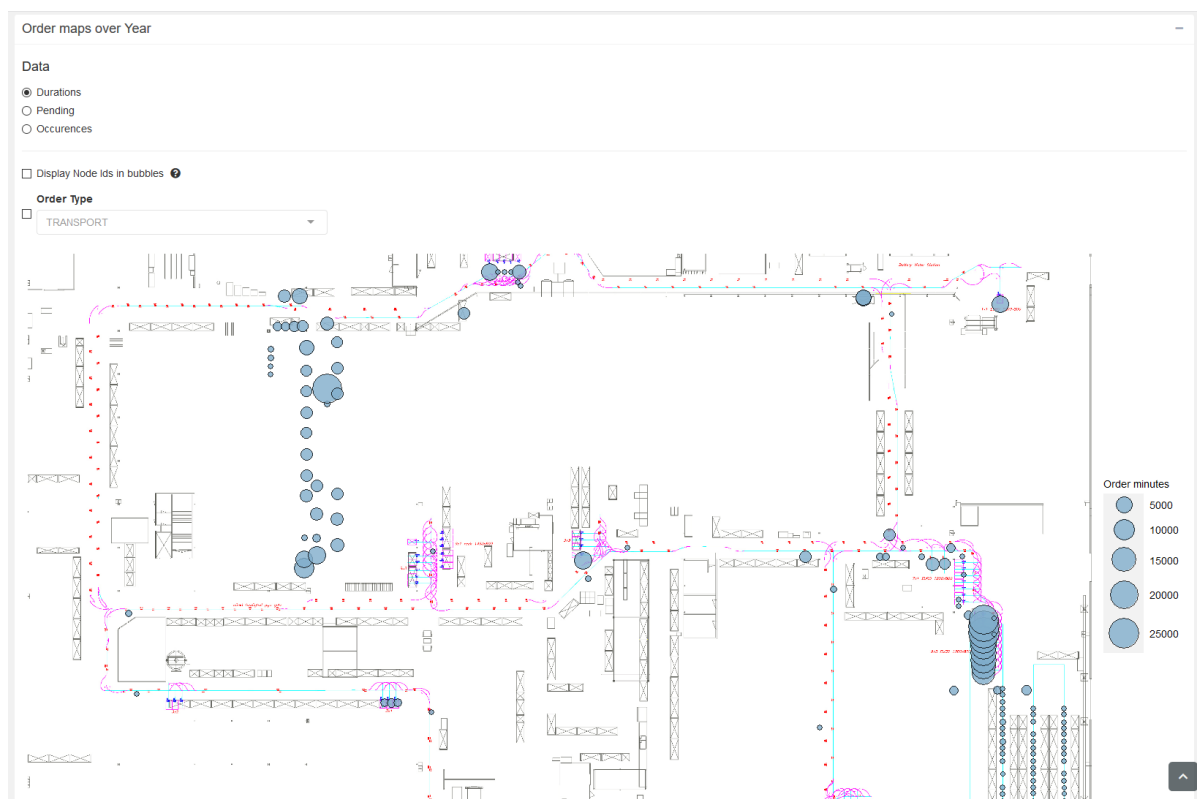


Fig 25, Orders Bubble Chart.

Occurrences presenteras med hjälp av en densitetskarta där varje order finns plottad som en punkt i layouten. I densitetskartan delas layouten upp i mindre areor eller så kallade "bins". Fler bins resulterar i en mer finfördelad uppdelning och en högre upplösning. Densiteten av ordrar inom varje bin visas genom en färgskala. Figuren nedan föreställer ett urklipp av två densitetskartor över samma område i layouten men med olika bin-värden.



Fig 26, Två urklipp ur densitetskartor över samma område.
Högra bilden har ett högre bin-värde och således en högre upplösning.

5.6 Alerts-sidan

Alerts-sidan följer samma mönster som för Utilization och Orders. Användaren kan ställa in parametrar och filter i inmatnings-boxen och till höger visas infoboxar med information om mängden alerts och längsta respektive genomsnittliga alert-tiden. Majoriteten av graferna följer samma struktur som i Orders, både mängden och tidsåtgången för varje alert finns representerad i varsin graf. Figurer för hela sidan finns nedanför.

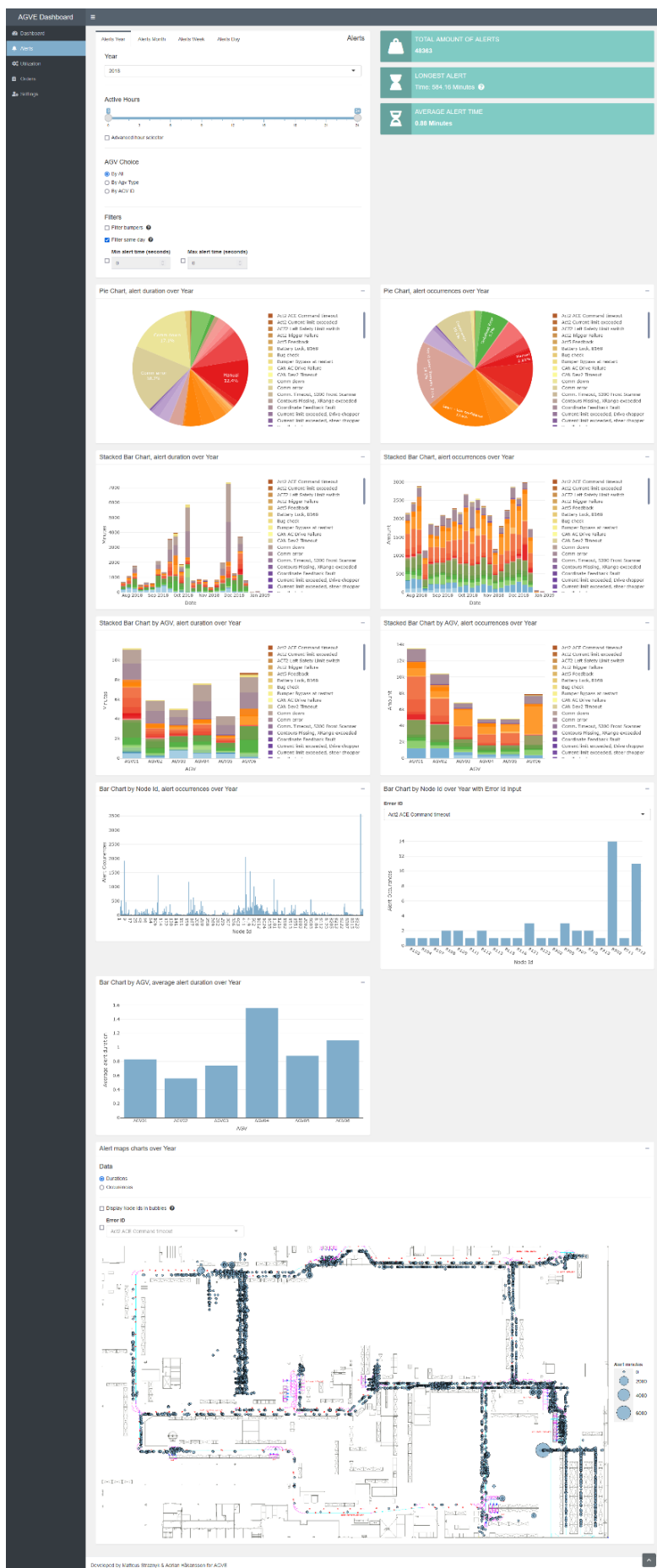


Fig 27, Alerts-sidan.

Visar vänster till höger, uppifrån och ner:

Användar inmatnings boxen,
Alerts total amount info box,
Alerts longest alert info box,
Alerts average time info box,
Alerts duration cirkeldiagram,
Alerts amount cirkeldiagram.

Alerts duration staplat stapeldiagram
över tid,

Alerts amount staplat stapeldiagram
över tid,

Alerts duration staplat stapeldiagram
per AGV,

Alerts amount staplat stapeldiagram per
AGV.

Alerts stapeldiagram per Node,

Alerts stapeldiagram per Node med
Alert Id inmatning.

Alerts average stapeldiagram per AGV,

Alerts bubble/density map.

Inmatnings-boxen för Alerts-sidan är annorlunda än de andra sidorna genom att ha en filter-sektion. Den innehåller fyra extra inställningar för att filtrera alert-statistiken, *Filter bumpers*, *Filter same day*, *min alert time* samt *max alert time*. Första inställningen *Filter bumpers* tar bort bumper-alerts om AGV:en står stilla eftersom dessa alerts sker ofta men är vanligtvis temporära och inte lika kritiska som andra alerts. Andra inställningen *Filter same day* tar bort alerts som varar längre än en dag. Tredje och fjärde inställningarna hör ihop med varandra genom att de båda filtrerar bort alerts som inte har en viss min eller max varaktighet (duration).

The screenshot displays the 'Alerts' configuration page. At the top, there are tabs for 'Alerts Year', 'Alerts Month', 'Alerts Week', and 'Alerts Day', with 'Alerts' selected on the right. Below the tabs is a 'Year' dropdown menu set to '2018'. The 'Active Hours' section features a horizontal timeline from 0 to 24 hours, with a blue bar indicating the active period from 7 to 20. Below this is an unchecked checkbox for 'Advanced hour selector'. The 'AGV Choice' section has three radio button options: 'By All' (selected), 'By Agv Type', and 'By AGV ID'. The 'Filters' section contains two checkboxes: 'Filter bumpers' (unchecked) and 'Filter same day' (checked). At the bottom, there are two input fields for 'Min alert time (seconds)' and 'Max alert time (seconds)', both set to '0' and accompanied by unchecked checkboxes.

Fig 28, Alerts använder inmatnings boxen.

I cirkeldiagrammen för Alerts (*Alert duration* och *Alert amount*) visas fördelningen mellan olika alerts både sett till tidsåtgång och antal. Användaren kan genom att jämföra graferna få en uppfattning om vilka alerts som är vanligast och vilka alerts som tar upp längst tid.

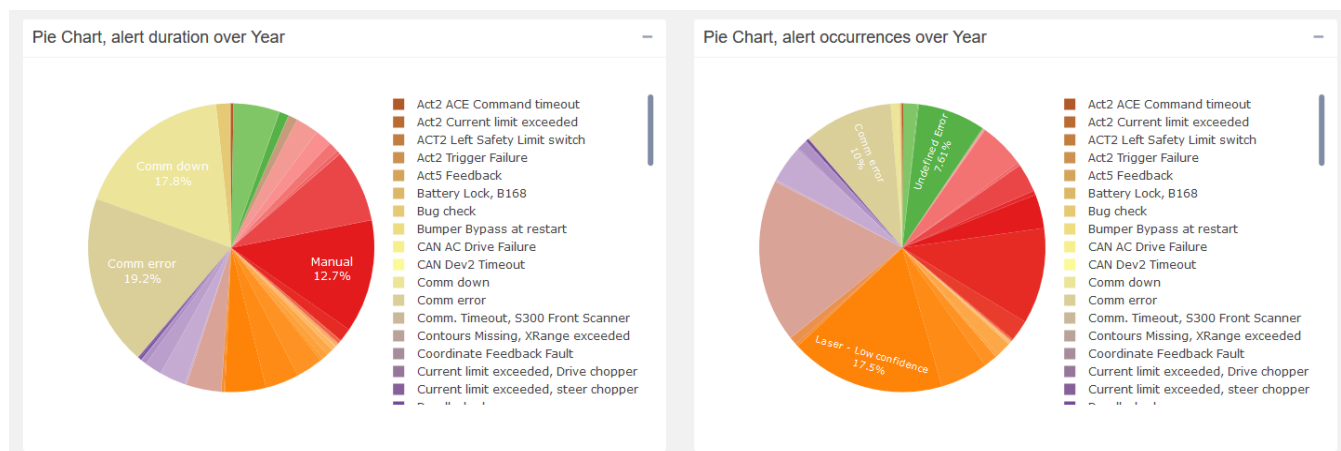


Fig 29, Alert duration och Alert amount cirkeldiagram.

De staplade stapeldiagrammen Alert duration över tid och Alert amount över tid visar fördelningen av alerts över det valda tidsspannet. I figuren nedan är tidsspannet inställt till en vecka och varje stapel representerar därför en dag. De här graferna kan hjälpa användaren att identifiera när en viss alert inträffade.

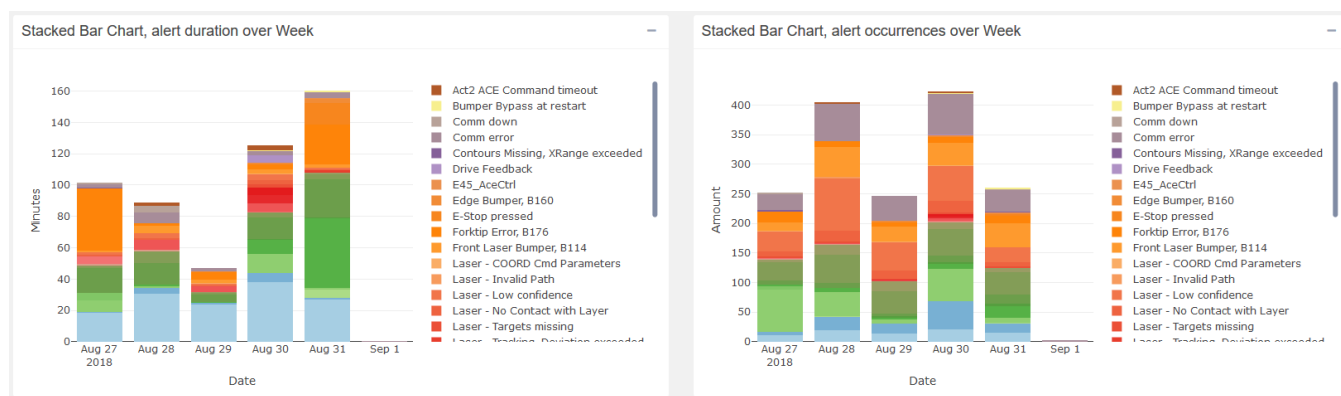


Fig 30, Alert duration över tid och Alert amount över tid staplat stapeldiagram

I Alert duration per AGV och Alert amount per AGV är alla alerts fördelade efter vilken AGV eller AGV-Type de uppstått hos. Här kan användaren identifiera om det finns en korrelation mellan en viss alert och en viss AGV.

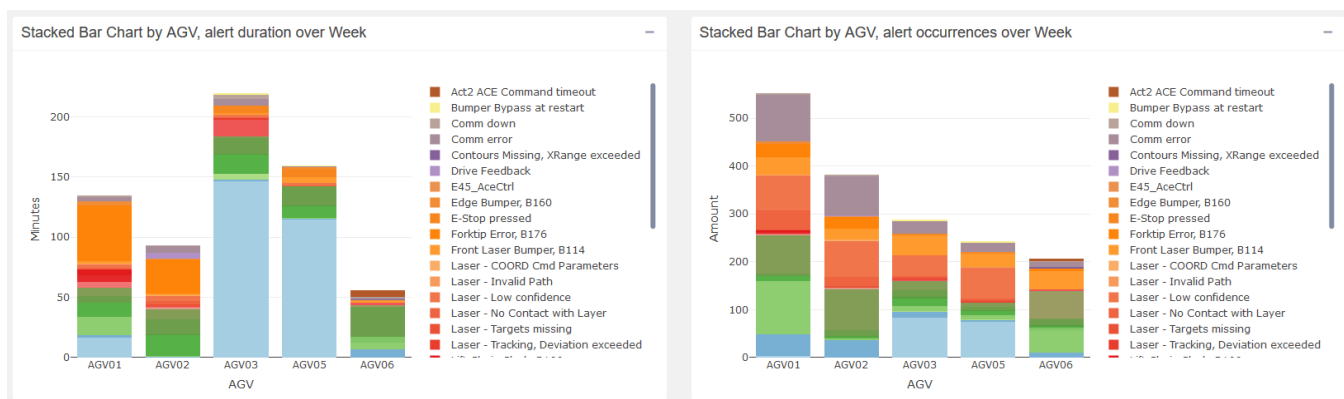


Fig 31, Alert duration per AGV och Alert amount per AGV staplat stapeldiagram.

Graferna Alert amount per Node och Alert amount per Node med alert Id-inmatning visar det totala antalet förekommande alerts fördelat över de noder där de inträffat. Den vänstra grafen (Alert amount per Node) visar en helhetsbild för alla typer av alerts i systemet under det valda tidsintervallet medan den högra grafen (Alert amount per Node med alert Id-inmatning) låter användaren ange en specifik alert-typ.

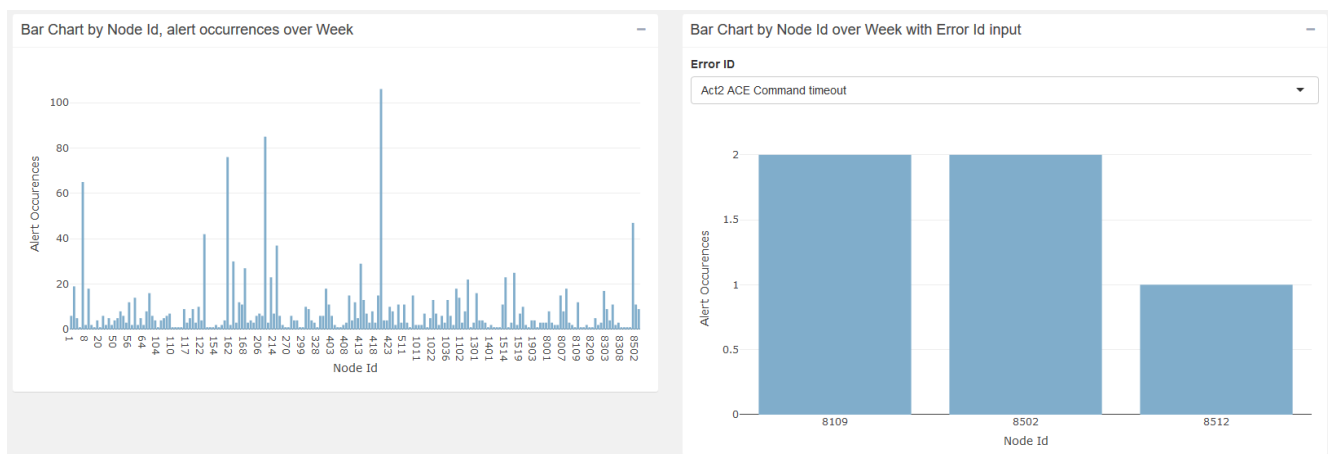


Fig 32, Alert amount per Node och Alert amount per Node med Alert Id inmatning.

Stapeldiagrammet Alerts average per AGV liknar grafen för Alerts duration per AGV men y-axeln visar den genomsnittliga tidsåtgången per alert istället för den totala tidsåtgången. Här kan användaren snabbt se om en AGV utmärker sig utan att behöva jämföra tidsåtgång och antal själv i graferna Alert duration per AGV och Alert amount per AGV ovan.

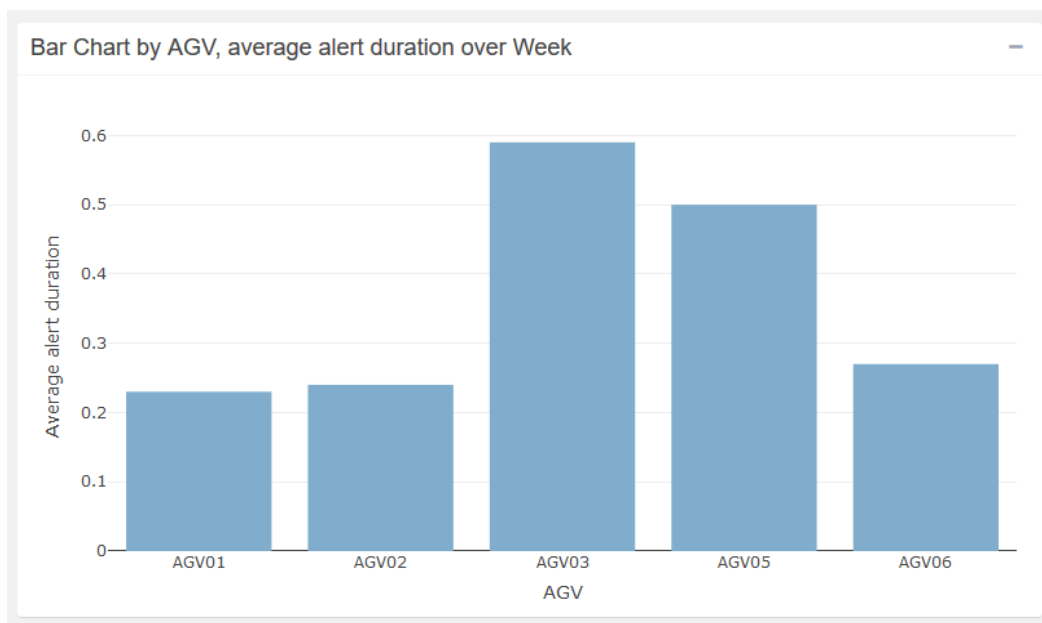


Fig 33, Alerts average stapeldiagram per AGV.

Precis som för ordrar presenteras också alerts i heatmaps i form av ett bubbeldiagram för tidsåtgången och en densitetskarta för antalet alerts. Användaren kan också välja att enbart se alerts av en specifik typ. Här kan användaren ta reda på om det finns en korrelation mellan en viss typ av alert och en plats i anläggningen.

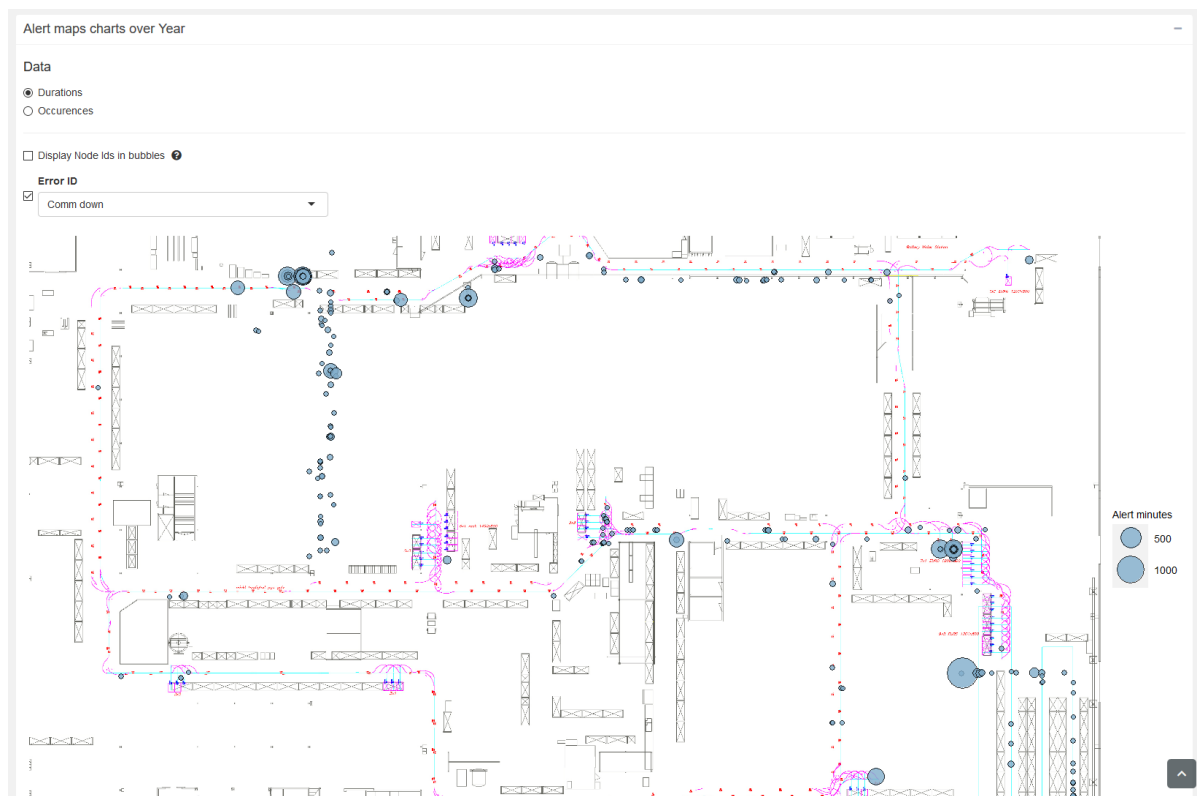


Fig 34, Alerts bubble/density map.

5.7 Settings-sidan

Settings-sidan har inställningar och standardvärden för webbapplikationen. Användaren går in på sidan om de behöver ändra layout-bilden och konfigurera den, eller sätta standard tim och veckodags-värden för hela applikationen. Värden som ställs in på den här sidan sparas globalt i applikationen och kommer användas i alla nästkommande sessioner. Se figuren nedan för hela Settings-sidan.

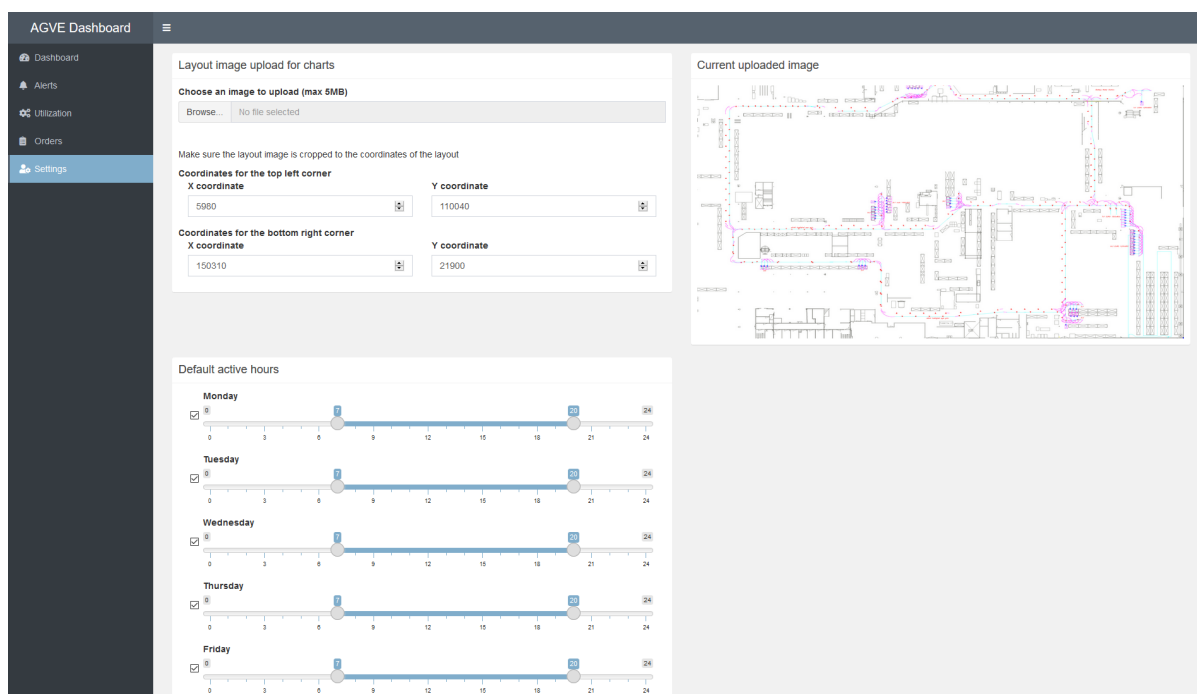


Fig 35, Settings-sidan.

Visar vänster till höger, uppifrån och ner:
Bilduppladdning och koordinat konfiguration,
förhandsvisning av den uppladdade bilden,
standard tim och veckodag inmatning.

Bilduppladdning och koordinatkonfigurations-boxen har två delar. I den första kan användaren bläddra sin lokala dator och ladda upp en bild till webbapplikationen. I den andra kan användaren konfigurera koordinaterna för bilden så att de matchar de interna koordinaterna som används i styrsystemet och statistikdatabasen. Se figuren nedan.

Layout image upload for charts

Choose an image to upload (max 5MB)

Browse... No file selected

Make sure the layout image is cropped to the coordinates of the layout

Coordinates for the top left corner

X coordinate: 0 Y coordinate: 0

Coordinates for the bottom right corner

X coordinate: 0 Y coordinate: 0

Fig 36, Bilduppladdning och koordinatkonfigurations-boxen.

Förhandsvisning av den uppladdade bilden, boxen gör vad den heter, den visar för användaren den nuvarande uppladdade bilden på webbapplikationen.



Fig 37, Förhandsvisning av den uppladdade bilden.

Standard tim och veckodags-inmatning tillåter användaren att ändra standard tidsspännet per veckodag. Varje veckodag har en slider med min och maxvärde för 24 timmar, och en checkmark för att helt och hållet sätta på eller stänga av hela dagen. Endast data inom de valda dagarna och intervallen kommer att presenteras genom hela applikationen.

Default active hours

Day	Active	Start	End
Monday	<input checked="" type="checkbox"/>	7	18
Tuesday	<input checked="" type="checkbox"/>	7	18
Wednesday	<input checked="" type="checkbox"/>	7	18
Thursday	<input checked="" type="checkbox"/>	7	18
Friday	<input checked="" type="checkbox"/>	7	16
Saturday	<input type="checkbox"/>	7	20
Sunday	<input type="checkbox"/>	7	20

*Fig 38, Standard tim och veckodag inmatning.
Med varje veckodag radade efter varandra
med en checkmark och slider.*

5.8 User Stories

Nedan följer lösningar för de user stories som sattes upp för projektet.

User story 1:

Hos ett företag har en alert på en AGV inträffat dagen innan, detta gjorde att AGV:en inte kunde arbeta. En chef ska nu ta reda på vilken alert som inträffat, på vilken AGV, när och hur det har påverkat antalet ordrar avslutade för den AGV:en och AGV-Typen.

Chefen går in på AGVE Dashboard-webbsidan för att få en närmare blick på hur det har påverkat AGV-systemet. När de gått in på webbsidan hamnar de på dashboard-sidan där de ser informationen att antalet ordrar dagen innan är liknande alla andra dagar, alltså inga ändringar på den fronten. De ser också att dagen innan fanns det en alert som varade 214 minuter, som har varit den längsta alerten den senaste veckan. När de svävar muspekaren över tooltip-ikonen får de informationen att alerten inträffade AGV nr 2 och alerten var en "Laser - target missing". De går sedan in på order-tabben för att ta reda på hur händelsen har påverkat antalet ordrar gjorda. Inne på tabben sorterar de efter veckor och väljer den nuvarande veckan och sorterar efter AGV nr 2, då ser de på stapeldiagrammet över veckan att vanligtvis arbetar AGV:en i 360 till 480 minuter men dagen innan så var den siffran ner till 100 minuter. Sedan sorterar de efter den AGV-typen som AGV:en tillhör, där ser de också att AGV:en har tillfört mindre jämfört med de andra, trots detta så har den totala antalet ordrar inte minskat, detta betyder att de andra AGV:erna i typen har tagit upp slacken eftersom systemet är uppbyggt med redundans.

User story 2:

En chef på ett företag vet att julen närmar sig och att antalet sålda produkter kommer att öka. Chefen vill ta reda på hur deras AGV-system har påverkats under samma period de föregående åren och hur trenden ser ut.

Chefen går in på AGVE Dashboard-webbsidan och går in på Orders-sidan, där väljer de det föregående året och tar alla timmar på dygnet under hela veckan. När de scrollar ner kommer de till det staplade stapeldiagrammet över tid och kontrollerar mängden ordrar som blir avslutade. Där ser de att i början av december ökar mängden ordrar jämfört med resten av året och ökningen varar tills nyår. När de gör samma sak för förföra året ser de något liknande. De drar slutsatsen att antalet ordrar ökar med cirka 30%-50% under perioden.

User story 3:

En chef på ett företag har fått höra från arbetarna på lagret att en av AGV:erna har strulat extra mycket vid en av arbetsstationerna. De vill nu gå och undersöka om detta är sant och i så fall vad problemet kan vara och var det händer.

Efter de går in på AGVE Dashboard-sidan går de in på Alerts-sidan, där väljer de vecko tabben och väljer den föregående veckan, de vanliga arbetstimmarna och alla AGV:er. När de scrollar ner och observerar det staplade stapeldiagrammet per AGV framkommer det att

AGV nr 4 har haft betydligt många fler alerts än de andra AGV:erna, så som arbetarna berättade om. Chefen går då tillbaka och väljer bara AGV 4 från användarinmatningen och scrollar ner till bubbeldiagrammet för att undersöka var AGV:en har haft problemen. Efter en snabb titt på diagrammet blir det uppenbart att en av bubblorna är mycket större än de andra. När de sätter på Node Id-texten syns det att bubblan tillhör Node 167, och det är där problemet förekommer.

6. Diskussion/Slutsats

6.1 Sammanfattning

Projektet omfattade ett arbete som började med ett nytt programmeringsspråk och miljö, samt ett främmande företag, styrsystem och databas. Under en period av fyra månader har en produkt utvecklats som i dagsläget är användbar och kravuppfyllande. Applikationen har grafer för statistiken som är användarvänliga och underlättar analys och felsökning av AGV-systemet. Med detta har projektet uppfyllt syftet och målet. Det finns också stora vidareutvecklingsmöjligheter för att göra applikationen till en potentiellt säljbar produkt.

Utvecklingen och metoden för framtagning av graferna anser vi har fungerat bra. R och Shiny:s verktyg för visualisering och grafer har varit väl anpassade för projektet. Något som hade kunnat förbättras är framtagningen av user stories. I början av projektet var inte dessa konkret definierade vilket resulterade i att mer tid fick ägnas åt att försöka hitta relevant information och sätt att presentera den på.

6.2 Användning

Applikationen kommer att kunna användas för:

- Intern testning och utveckling av AGV:er.
- Teknisk support hos kunder. Genom att en AGVE tekniker åker ut till kund och felsöker problem med hjälp av applikationen.
- Möjligtvis som säljbar programvara till kunder efter vidareutveckling och förbättring.

6.3 Problem

Under utvecklingen stötte vi på ett antal problem och utmaningar. Här följer beskrivningar av ett antal av dessa, samt hur vi hanterat dem.

- För bubbel- och densitetsgraferna behövs en bakgrundsbild av anläggningens layout. Men Plotly-paketet som används för de interaktiva graferna i resten av applikationen har inte denna funktionalitet. Just för dessa grafer har vi därför fått använda oss av ggplot2-grafer istället som inte är interaktiva.
- Tidigt i projektet märkte vi att vi inte kunde återanvända komponenter på flera ställen. Ett tag in i projektet användes kodduplikation för detta men med hjälp av Modules från Shiny kunde detta senare åtgärdas.
- När vi skulle lägga till tooltips i applikationen hittade vi ett paket som heter Tippy som gjorde detta. Men några veckor senare när vi ville göra tooltipsen dynamiska beroende på användarinmatning, märkte vi att det inte fungerade. Vi fick därför undersöka och hitta ett nytt paket, och det var så vi hittade ShinyBS paketet.

- Efter att vi bestämde oss för att använda staplat stapeldiagram i applikationen märkte vi att datan vi fick från queryn sen tidigare inte skulle fungera. Detta eftersom datan vi hämtade hade en rad per datatyp i tabellen, och vi skulle behöva en rad med alla datatyper som kolumner istället. Med lite undersökning inom SQL hittade vi en funktion som heter Pivot [34]. Pivot roterar värden från en kolumn till flera kolumner som output och eftersom systemet behöver vara dynamiskt behövdes en dynamisk pivot göras. Detta betyder att de kolumnerna som datan skulle roteras till behövdes för-hämtas och förberedas för att infogas in i queryn.
- Ett problem som framkom under projektets gång var att i takt med att vi lärde oss nya tekniker inom R upptäcktes förbättringar som gjorde att kodbasen behövde refaktoreras flera gånger.

6.4 Etik

Under projektets gång har vi funderat på ett etiskt problem med applikationen. Det är att de diagram som visualiserar statistiken inte får vara inkorrekta. Detta är eftersom applikationen kommer användas för analys och felsökning, och om statistiken blir visad på ett inkorrekt sätt kan AGVE eller en av deras kunder dra fel slutsatser och göra felaktiga beslut baserat på felaktig data.

Ett sätt vi motverkade problemet var genom att testa applikationen så att den visualiserar statistiken på rätt sätt genom att jämföra med den aktuella datan på databasen.

6.5 Förbättringar och vidareutveckling

Nedan följer ett antal förbättrings ideer och vidareutvecklingsmöjligheter i punktform:

- Öka interaktiviteten av graferna genom att hantera click event när man klickar på en graf, som till exempel man klickar på stapel för en specifik vecka som sedan tar dig till en sida med mer informationen för just den veckan.
- Uppdatera paketet som skapar Dashboard layouten från shinydashboard till bs4dash 2.0.0 när den kommer ut till CRAN. Eftersom shinydashboard använder sig av adminlte 2 och bootstrap 3 jämfört med bs4dash som använder sig av adminlte 3 och bootstrap 4.
- Öka effektiviteten på applikationen så att den är mer responsiv och snabbare. Den stora flaskhalsen för applikationen i dagsläget är att många queries körs på flera stora tabeller.
- Funktionalitet för att jämföra olika grafer med varandra. Genom att graferna har en knapp i sina boxar som sparar graferna till en jämförelsesida. På den sidan kommer

graferna ligga bredvid varandra så att användaren enkelt kan jämföra dem med varandra. På sidan skulle det också gå att ta bort graferna från jämförelsen.

- Utveckla automatisk analys av statistiken genom att studera korrelationer och outliers, samt framställa de för användaren på ett begripligt sätt.
- Lägga till funktionalitet för användaren att generera rapportfiler i pdf eller annat format. Användaren ska kunna ange ett antal grafer med specifik inmatning som senare sammanställs i en rapport som kan laddas ner. För skapandet av rapporten ska det både finnas användarinterface och generering från fil som i till exempel JSON eller XML.

Källförteckning

[1]"Automated guided vehicles (AGVs) – AGV systems", AGVE, 2021. [Online]. Available: <https://www.agvegroup.com/>. [Accessed: 21- May- 2021].

[2]"Automated guided vehicle - Wikipedia", En.wikipedia.org, 2021. [Online]. Available: https://en.wikipedia.org/wiki/Automated_guided_vehicle. [Accessed: 21- May- 2021].

[3]"R: What is R?", R-project.org, 2021. [Online]. Available: <https://www.r-project.org/about.html>. [Accessed: 21- May- 2021].

[4] "About RStudio", Rstudio.com, 2021. [Online]. Available: <https://www.rstudio.com/about/>. [Accessed: 25- May- 2021].

[5] "nanxstats/awesome-shiny-extensions", GitHub, 2021. [Online]. Available: <https://github.com/nanxstats/awesome-shiny-extensions>. [Accessed: 25- May- 2021].

[6] "The Comprehensive R Archive Network", Cran.r-project.org, 2021. [Online]. Available: <https://cran.r-project.org/>. [Accessed: 25- May- 2021].

[7]"Shiny", Shiny.rstudio.com, 2021. [Online]. Available: <https://shiny.rstudio.com/>. [Accessed: 21- May- 2021].

[8]"Plotly R Graphing Library", Plotly.com, 2021. [Online]. Available: <https://plotly.com/r/>. [Accessed: 21- May- 2021].

[9]"Shiny Dashboard", *Rstudio.github.io*, 2021. [Online]. Available: <https://rstudio.github.io/shinydashboard/>. [Accessed: 21- May- 2021]

[10]"Add More AdminLTE2 Components to shinydashboard", *Rinterface.github.io*, 2021. [Online]. Available: <https://rinterface.github.io/shinydashboardPlus/>. [Accessed: 21- May- 2021]

[11]"Create Custom Bootstrap Themes to Use in Shiny", *Dreamrs.github.io*, 2021. [Online]. Available: <https://dreamrs.github.io/fresh/>. [Accessed: 21- May- 2021]

[12]"Databases using R", *Db.rstudio.com*, 2021. [Online]. Available: <https://db.rstudio.com/odbc>. [Accessed: 21- May- 2021]

[13]"A Grammar of Data Manipulation", *Dplyr.tidyverse.org*, 2021. [Online]. Available: <https://dplyr.tidyverse.org/>. [Accessed: 21- May- 2021]

- [14]"Create Elegant Data Visualisations Using the Grammar of Graphics", *Ggplot2.tidyverse.org*, 2021. [Online]. Available: <https://ggplot2.tidyverse.org/>. [Accessed: 21- May- 2021]
- [15]"Plotly R Graphing Library", *Plotly.com*, 2021. [Online]. Available: <https://plotly.com/r/>. [Accessed: 21- May- 2021]
- [16]D. Attali, "shinyjs", *Deanattali.com*, 2021. [Online]. Available: <https://deanattali.com/shinyjs/>. [Accessed: 21- May- 2021]
- [17]F. package, R. docs, R. browser and R. Palettes, "RColorBrewer: ColorBrewer Palettes version 1.1-2 from CRAN", *Rdrr.io*, 2021. [Online]. Available: <https://rdrr.io/cran/RColorBrewer/>. [Accessed: 21- May- 2021]
- [18]"jpeg - Read, write and plot JPEG image files in R - RForge.net", *Rforge.net*, 2021. [Online]. Available: <http://www.rforge.net/jpeg/>. [Accessed: 21- May- 2021]
- [19]"shinyBS", *Ebailey78.github.io*, 2021. [Online]. Available: <https://ebailey78.github.io/shinyBS/>. [Accessed: 21- May- 2021]
- [20]"daattali/shinycssloaders", *GitHub*, 2021. [Online]. Available: <https://github.com/daattali/shinycssloaders>. [Accessed: 21- May- 2021]
- [21]"DT: An R interface to the DataTables library", *Rstudio.github.io*, 2021. [Online]. Available: <https://rstudio.github.io/DT/>. [Accessed: 21- May- 2021]
- [22]"Make Dealing with Dates a Little Easier", *Lubridate.tidyverse.org*, 2021. [Online]. Available: <https://lubridate.tidyverse.org/>. [Accessed: 21- May- 2021]
- [23]"Shiny - Modularizing Shiny app code", *Shiny.rstudio.com*, 2021. [Online]. Available: <https://shiny.rstudio.com/articles/modules.html>. [Accessed: 21- May- 2021]
- [24]"Databases using R", *Db.rstudio.com*, 2021. [Online]. Available: <https://db.rstudio.com/best-practices/run-queries-safely/>. [Accessed: 21- May- 2021].
- [25]"Shiny - Reactivity - An overview", *Shiny.rstudio.com*, 2021. [Online]. Available: <https://shiny.rstudio.com/articles/reactivity-overview.html>. [Accessed: 21- May- 2021].
- [26]"Pie Charts", *Plotly.com*, 2021. [Online]. Available: <https://plotly.com/r/pie-charts/>. [Accessed: 21- May- 2021].
- [27]"Bar Charts", *Plotly.com*, 2021. [Online]. Available: <https://plotly.com/r/bar-charts/>. [Accessed: 21- May- 2021].

[28]"ColorBrewer: Color Advice for Maps", *Colorbrewer2.org*, 2021. [Online]. Available: <https://colorbrewer2.org/>. [Accessed: 21- May- 2021]

[29]"Points — geom_point", *Ggplot2.tidyverse.org*, 2021. [Online]. Available: https://ggplot2.tidyverse.org/reference/geom_point.html. [Accessed: 21- May- 2021]

[30]"Heatmap of 2d bin counts — geom_bin2d", *Ggplot2.tidyverse.org*, 2021. [Online]. Available: https://ggplot2.tidyverse.org/reference/geom_bin2d.html. [Accessed: 21- May- 2021]

[31]"shinyapps.io", *Shinyapps.io*, 2021. [Online]. Available: <https://www.shinyapps.io/>. [Accessed: 08- Jun- 2021]

[32]"Shiny Server", *Rstudio.com*, 2021. [Online]. Available: <https://www.rstudio.com/products/shiny/shiny-server/>. [Accessed: 08- Jun- 2021]

[33]"ShinyProxy", *Shinyproxy.io*, 2021. [Online]. Available: <https://www.shinyproxy.io/>. [Accessed: 08- Jun- 2021]

[34]"Using PIVOT and UNPIVOT - SQL Server", *Docs.microsoft.com*, 2021. [Online]. Available: <https://docs.microsoft.com/en-us/sql/t-sql/queries/from-using-pivot-and-unpivot>. [Accessed: 21- May- 2021]

[35]"A4 - AGVE", *AGVE*, 2021. [Online]. Available: https://www.agvegroup.com/portfolio_item/a4/. [Accessed: 08- Jun- 2021]