

# Human Health and Activity Laboratory (H2Al)

## UWB-navigering av robot - Grupp 12

---

**Kurs: D0020E, Projekt i datateknik, Lp2-3, H22**

**Författare:**

Furhoff, Hannes	hanfur-0@student.ltu.se	010929-4710
Kebede, Mebaselassie Kidane	mebkeb-0@student.ltu.se	990412-5912
Nord, Oscar	oscnor-9@student.ltu.se	950124-6236
Valdivia Vargas, Andrés	andval-6@student.ltu.se	971116-1894

**Projekt Handledare:**

Kåre Synnes   kare.synnes@ltu.se

20 mars 2023



## Sammanfattning

Detta projekt syftar till att kombinera en så kallad *Pepper-robot* med UWB-positioneringssystem med hjälp av Widefind kommunikation inom äldreomsorg genom att använda en hållbar högteknologiska lösning på personalbrist. Roboten används nya tekniker för att kunna navigera sig till den önskad plats där personen befinner sig för att hjälpa till vid t.ex olycka. Projektet utfördes den 7:e november 2022 på Luleå Tekniska Universitet under handledning av Professor Kåre Synnes.

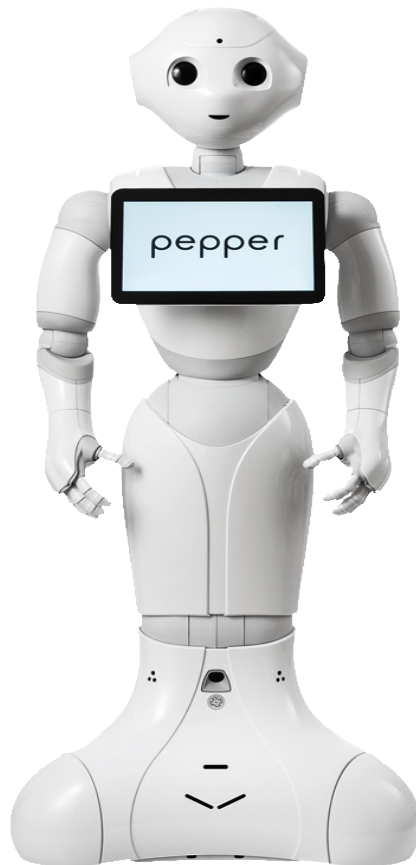
# Innehåll

<b>1</b>	<b>Inledning</b>	<b>1</b>
1.1	Bakgrund . . . . .	2
1.2	Problembeskrivning . . . . .	2
1.3	Uppgift och mål . . . . .	3
<b>2</b>	<b>Systemdesign</b>	<b>4</b>
2.1	UWB positioneringssystem . . . . .	5
2.2	GUI fall handler . . . . .	6
2.3	GUI Fall Handler . . . . .	7
<b>3</b>	<b>Resultat</b>	<b>8</b>
3.1	Sprint backlog . . . . .	8
3.2	Förbättring och framtida tillägg . . . . .	8
3.3	Länkar . . . . .	9

# 1 Inledning

Den vanligaste futuristiska tekniken som skildras inom science fiction kretsar kring robotar. De flesta människor idag föreställer sig robotar som används för att tillverka fordon, fungera som japanska kontorsreceptionister, städa hus automatiskt eller till och med flyga drönare. I nuläget de potentiella fördelarna för smarta robotar som verkar kunna utföra uppgifter mer effektivt än människor verkar dock obegränsade.

I denna projekt används en semi-humanoid robot som heter *Pepper* 1 som kan känna igen känslor baserat på upptäckt och analys av ansiktsuttryck och rösttoner. Ursprungligen tillverkades och marknadsfördes av det franska företaget **Aldebaran Robotics** 2014, senare togs över av det japanska företaget **Softbank**. [wiki]



Figur 1: Robot-Pepper.

## 1.1 Bakgrund

Idén med att ha en robot som hjälpmedel har funnits länge, i början bestod de av väldigt enkla människo- och djurliknande robotar som kunde utföra enkla rörelser [6]. På 60-talet började det dyka upp robotar för industrifabrik som kunde genomföra enkla moment [9]. Men autonoma robotar för hemmabruk började inte ta fart på riktigt förrän 1997 då Electrolux lanserade den första robotdammsugaren [4].

Nu har autonoma robotar blivit allt vanligare att se, några exempel på robotar som redan har börjat dyka upp är: Robotdammsugare har enligt iRobot år 2016 tagit över 20% av all försäljning av dammsugare [5]. Foodora testar självkörande leveransrobotar [2] och reklam och informationsrobotar i kinesiska köpcentrum [7].

På Luleå Tekniska Universitet så kan man hitta Aktivitetslaboratoriet [1] som bedriver forskning inom sensorbaserade system för att skapa nya sätt att hjälpa och stödja människor med olika funktionsnedsättningar. Ett av dessa system är ett positionssystem som använder sig av UWB vilket är en teknologi som kan positionera människor och föremål med hög noggrannhet [8]. Under detta projekt har ett UWB-system utvecklats av ZeroKey används. Utöver UWB-systemet kan man också hitta tre stycken robotar i laboratoriet, den mest avancerade av dessa är den roboten Pepper [3].

ZeroKey är ett Kanadensiskt teknikföretag som har utvecklat ett positioneringssystem med millimeter precision som använder sig av UWB. [11] Deras positionssystem fungerar genom att placera ut så kallade ankarnoder i taket som skickar ut signaler för att hitta de rörliga noderna. När de fasta ankarnoderna får kontakt med de rörliga noderna skickas ett meddelande som innehåller data med bland annat position, orientering av noden, batterinivå, nodens MAC-adress, vilken zon den befinner sig i med mera. [10] Detta ger ZeroKey möjligheten att i realtid kunna se exakta positionen i X, Y och Z led av personen eller föremålet som har en rörlig nod på sig.

Pepper är en humanoid robot designad för att kunna interagera med människor på ett mänskligt sätt genom att kunna läsa av känslor och kroppsspråk. Detta har lett till att Pepper används som bland annat receptionister då ansiktigenkänningsfunktionen automatiskt identifiera gästerna och skicka ett meddelande till rätt person och eskortera gästen till mötesplatsen. Pepper används även av skolor för att ge studenterna en första inblick i programmering av en humanoid robot.

## 1.2 Problembeskrivning

De typiska autonoma robotarna i moderna hushåll och miljöer utför enkla uppgifter; Köra mat, dammsuga golv, med mera. Det blir då naturligt att undra hur denna typ av robot kan hjälpa människan utöver endast dessa enkla uppgifter. En robot som sköter olika vardagssysslor och hjälper till i hemmet är en lockande idé. I detta projekt kollar vi på en liknande applikation inom hälso- och sjukvården.

Den del av sektorn som detta projekt inriktar sig på relaterar mot assistans till rörelsehindrade och äldre. Mer specifikt, att genom autonoma robotar ge assistans till personer som råkat ut för en fallolycka. Fallolyckor är ett vanligt problem, speciellt hos äldre, som kan ha allvarliga konsekvenser. Även i fall med lindriga skador, kan det vara svårt för personer att ta sig upp igen och därmed krävs insats från vårdpersonal. Vårt projekt utgår ifrån att hjälpa personer som vart med i en fallolycka genom att snabbt kontakta vårdpersonal.

I en mindre grad kommer vi också täcka personlig kontakt. Det finns redan en marknad för liknande notifieringssystem, tex. Minifinder Nano. Därför blir en viktig del av arbetet att bilda en mer personlig koppling mellan roboten och klient. Detta kommer göras genom tekniker såsom robotens tal- och konversationsförmåga, mänskliga utseende och rörelser, osv.

### 1.3 Uppgift och mål

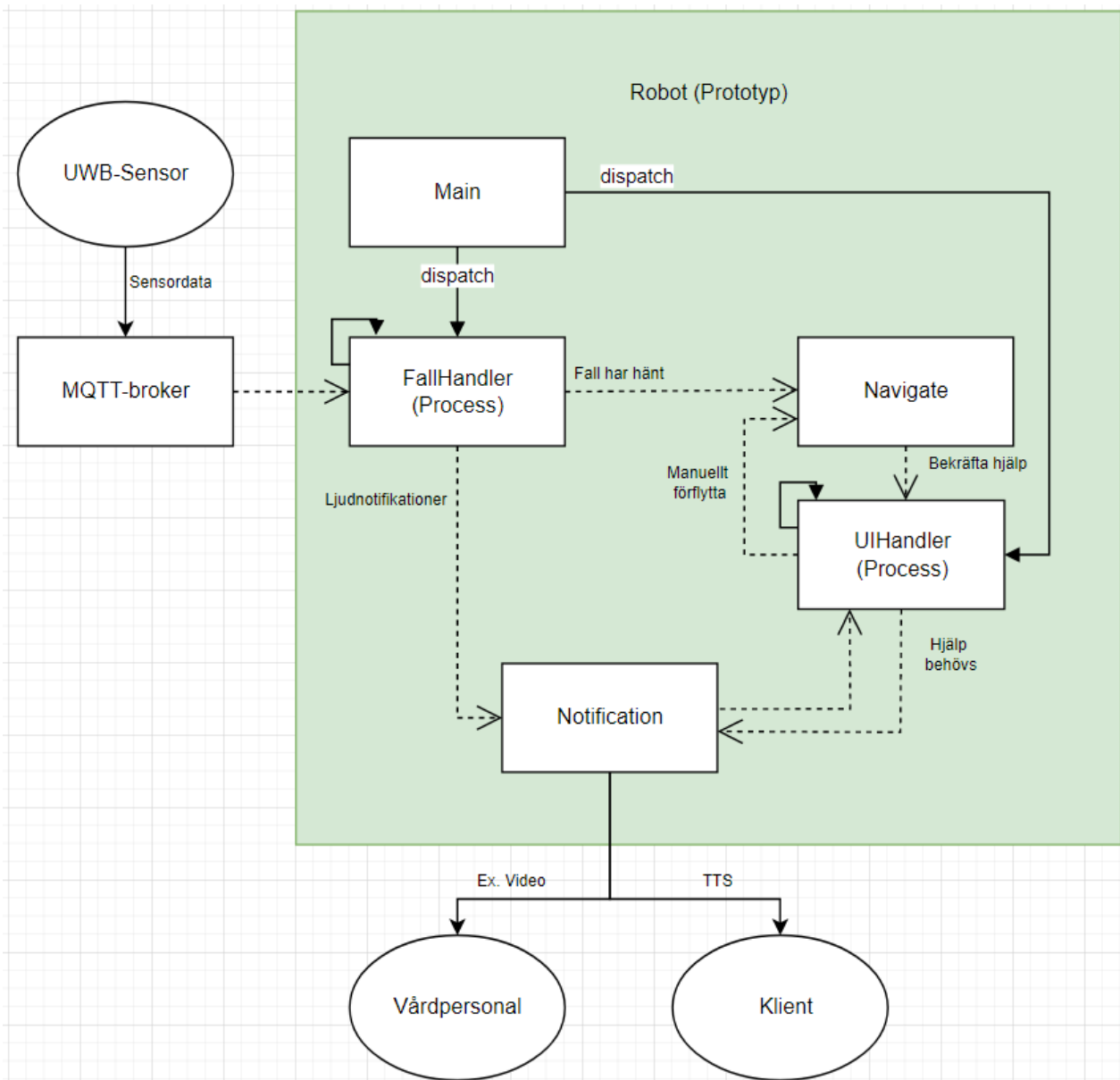
Uppgiften att använda roboten *Pepper* i sjukvården för att hjälpa äldre individer innebär att integrera roboten i deras vådrutiner med målet att ge bland annat känslomässigt stöd, förbättra deras övergripande välbefinnande och höja deras livskvalitet när familjemedlemmar eller vårdgivare inte är närvarande. Användningen av *Pepper* kommer bidra till att minska bördan för vårdgivare och vårdpersonal genom att tillhandahålla en ytterligare källa till stöd och hjälp till individerna.

Utöver en mer personlig närvaro och känslomässigt stöd ska roboten också vara tillgänglig att snabbt och effektivt kunna assistera vid fallolyckor. När en person har ramlat kan de vara tex. förvirrade eller allvarligt skadade, och inte alltid har förmågan att kontakta hjälp själva. Vår robot ska då kunna köra fram till den skadade personen och kontakta sjukvårdspersonal åt de.

Ytterligare ett krav för systemet är stabilitet. På grund av appliceringsområdet så är det viktigt att systemets alla komponenter kan fungera i princip felfritt, eller kunna tydligt notifiera i god tid om någonting går snett. På grund av de många olika implementationssvårigheterna kan det här målet kräva utförliga tester av komponenter och system. Ett verktyg för att uppnå detta är att isolera systemets moduler, och därmed ha ett system som blir mindre komplicerat och därmed lättare att identifiera och fixa buggar och fel.

Detta görs genom att programmera roboten att delta i konversation med den äldre individen. Är det möjligt att integrera ett UWB-positioneringssystem inomhus med en humanoid robot för att ge en fjärrassistans vid en fallolycka i hemmiljö? Svaret är ja, i detta projekt kommer roboten användas som hemassistent vid en fallolycka för att då koppla ihop den äldre individen med vårdpersonal genom video eller direktsamtal genom att använda en hållbar högteknologiska lösning.

## 2 Systemdesign



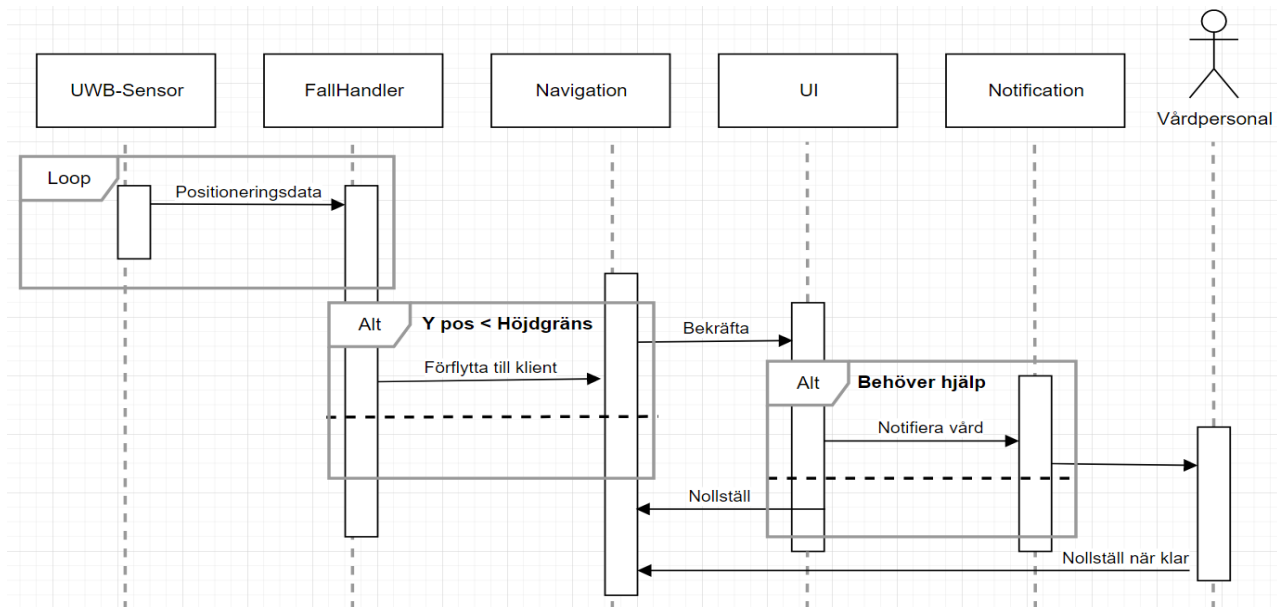
Figur 2: Statisk design: Moduldiagram.

Diagrammet i Figur 2 Visar den statiska designen för vårt system. Vi har valt att använda moduldiagram (UML) på grund av hur interaktionerna mellan modulerna ser ut. I vår prototyp fungerar varje modul som en svart låda"i relation till de andra modulerna, där minimal koppling mellan moduler är önskvärt. Varje moduls uppgift är isolerad, enkelt beskriven och enkelt förstådd, där arbetet i systemets utveckling är mestadels kopplad till implementationen. Därför kan specifika detaljer såsom funktioner och variabler bortses utan att systemet blir svårförstått.

Alla komponenter som är beskrivna i cirklar ska tolkas som yttre aktörer, till exempel sensorer eller människor. Rektanglar beskriver processer och moduler som är direkt kopplade till systemet, såsom MQTT-brokers och klasser. För varje pil som syftar på ett beroende (de streckade pilarna) så finns det text som antyder kontexten under vilket relationen kallas. Ytterligare ska alla moduler i den gröna rutan köras på Pepper-roboten via det inbyggda operativsystemet.

Systemets flöde beskrivs ytligt i detta diagram, men för en tydligare vy över de logiska operationernas sekvens under systemets exekvering så har vi valt att använda sekvensdiagram för den dynamiska designen (se Figur 3).

## 2.1 UWB positioneringssystem



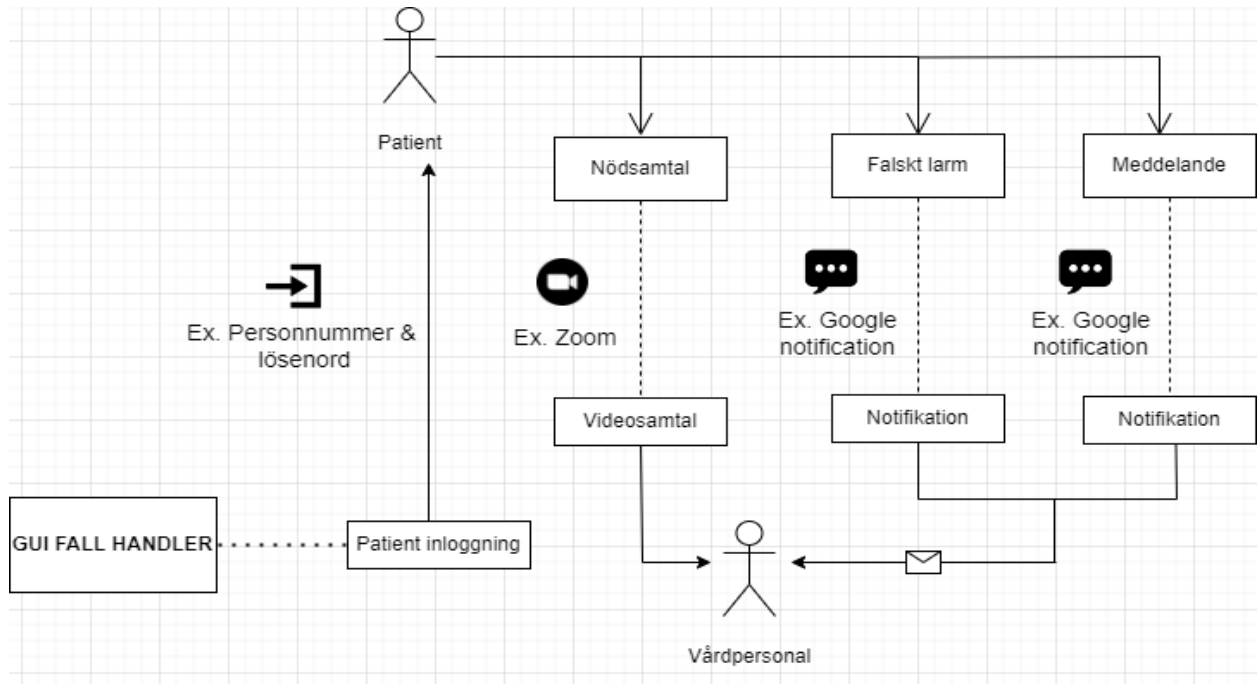
Figur 3: Dynamisk design: Sekvensdiagram.

Sekvensdiagrammet från Figur 3 läses från vänster till höger, från positioneringssensorn ("UWB-Sensor") till vårdpersonalen som ska hjälpa klienten. För boxar markerade Loop körs funktionen på insidan om och om så länge som systemet förblir operativt. För boxar markerade Älså gäller en if-sats där uttrycket är givet i övre delen av boxen till höger om textrutan (till exempel. "Y pos < Höjdgräns" från diagrammet). Om uttrycket är sant, följs pilen i övre halvan. Om uttrycket är falskt följs pilen i nedre halvan. Finns det ingen pil i en halva så antas funktionsflödet sluta exekvering där.

Valet av sekvensdiagram var gjort för att förtydliga hur systemet kommer fungera i praktik, där implementationsdetaljer såsom MQTT-broker och processhantering inte inkluderas för tydlighetens skull. Ett aktivitetsdiagram skulle också vara en lämplig modell för systemet av samma anledning, men vi valde sekvensdiagram då systemets interaktioner är bättre beskrivet som modul-till-modul snarare än aktivitet-till-aktivitet.



## 2.2 GUI fall handler



Figur 4: GUI fall handler dynamisk design.

Figuren 5 visar den övergripande koncepten om hur vår applikation kommer att fungera efter att ett fall har upptäckts. Det visar också en sekvens av steg som applikationen kommer att genomföra för att komma i kontakt med vårdpersonalen.

Applikationen kommer att ha en gränssnitt anpassat för en äldre patient och även för någon som inte har ett djupt IT kunskap eller förståelse. Detta gränssnitt kommer att vara en viktig del av applikationen eftersom vi vill att det ger användare en snabb och enkel åtkomst till all relevant information samt alternativ.

## 2.3 GUI Fall Handler

Softbank Android API är den primära källa för dokumentation som används för att styra en Pepper robotapplikation via det nya Android API:et som är tillgängligt från robotversion 2.9 och uppåt. Tyvärr var denna version av roboten gammal och byggd på en Android version som för närvarande körs i 5.0 kärnversionen från 2016, vilket gjorde svårt att integrera den med det nya Android API:et. Efter många försök då kom vi fram till två olika tillvägagångssätt för att lösa problemet och integrera en äldre version av robot med det nya Android API:et.

Eftersom robotens hjärna och den inbyggda surfplatta kommunicerar på olika sätt, det första sättet var att anropa ett *Python* skript som i sin tur kallade ett separat *html* samt *javascript* med alla funktionalitet direkt från robotens terminalen efter man `ssh:at` i med robotens egen IP adress och lösenord. Detta tillvägagångssätt fungerade i stort sett som vi ville tills vi kom fram till den delen där patienten behövde kommunicera med vårdpersonalen via ett video samtal. Denna version av roboten tillåter samt stödjer inte att öppna moderna applikationer i en ny flik som *Zoom* vilket används som kommunikationstjänst mellan dessa två individer i detta projekt.

Det andra tillvägagångssättet som fungerade till slut var att behålla vår användargränssnitt som är anpassat för de äldre patienterna och istället anropa det från terminalen. För att underlätta kommunikationen mellan virtuella maskiner och robotens inbyggda surfplatta integrerade vi ett *Zoom* web SDK med vår applikation genom att använda hemliga API-nycklar, slutpunkter och en värdserver i ett privat virtuellt nätverk LUDD-dust.

Vår applikation stödjer ett patient inloggningsida med en befintlig personnummer och lösenord som består av tre alternativ på huvudsidan, såsom *nödsamtal*, *falskt larm* och *meddelande*. Detta underlättar kommunikationen mellan dessa individer och vårdpersonalen. Eftersom robotens fall detektion funktionen är ännu inte fullständigt felfri, det kan hända att den inte upptäcker när ett fall inträffas. Genom att välja alternativet *meddelande* patienten kan notifiera vårdaren om att ett fall har inträffat och be om hjälp. Om det inte var ett fall men ändå upptäcker roboten något, kan de välja *falskt larm* alternativ och notifiera vårdpersonalen samt möjligtvis beordra roboten att köra till sin laddningsstation.

Huvudfunktionen i vår applikation är *nödsamtal*, där roboten körs fram till den plats där patienten befinner sig med hjälp av UWB-positioneringssystem och går med i ett *Zoom* möte via videosamtal, vilket ger patienten möjlighet att prata med vårdpersonalen om situationen.

I nuläget är applikationen ännu inte komplett, men den fungerar som ett prototyp demo i idéstadiet och testning och den erbjuder därmed utrymme för ytterligare förbättringar och tillägg i framtiden som beskrivs i 3.2.

## 3 Resultat

### 3.1 Sprint backlog

	Title	Priority	Size	Risk	Status	Expected Time
1	Task ID 1: Tillgång till robot	Viktig	Små	Låg	Färdig	3-4 Timmar
2	Task ID 2: Läs in data från sensorer	Hög	Små	Låg	Färdig	6 timmar
3	Task ID 3.1: Widefind-utskrift	Viktig	Medium	Medium	Färdig	8 timmar
4	Task ID 3.2: Förflytta robot	Hög	Medium	Låg	Färdig	10 timmar
5	Task ID 4.1: Notifikationer (TTS)	Medium	Små	Medium	Färdig	6 timmar
6	Task ID 4.2: Notifikationer (VC)	Hög	Stor	Hög	Färdig	4 timmar
7	Task ID 5.1: GUI design av användargränssnitt	Medium	Stor	Låg	Färdig	6 timmar
8	Task ID 5.2: GUI implementation	Hög	Stor	Hög	Färdig	10 timmar
9	Task ID 6: Kombinera och testa	Låg	Medium	Medium	Färdig	12 timmar

Figur 5: Sprint 1 och 2 backlogg.

### 3.2 Förbättring och framtida tillägg

Navigerings klassen använder just nu en funktion som placerar Pepper i riktning mot negativ X enligt ZeroKey's UWB koordinater. Denna funktion bygger på en av Peppers inbyggda positioneringsverktyg som är väldigt inkonsekvent, även under samma session så kan de koordinater för rotation ändras vilket koden saknar en funktion för att kontrollera och åtgärda automatiskt. Det finns ett flertal sätt att lösa detta problem, ett av dem är att använda orinteringen som skickas ut från ZeroKey.

En förbättring när det gäller applikationen är att ha ett specialiserade databas för varje patient, där det lagras deras personnummer, lösenord samt vård journal inklusive ett specialiserade *Zoom* möte med ett säkert nätverk och ett unikt Notifieringssystem till vårdpersonalen när det ett fall inträffas. Ett separat inloggning sida samt inställningar för vårdpersonalen bör implementeras där då kan följa status av patientenshälsa, fel anmäla roboten om det uppstår något fel och andra saker som kan fullborda applikation till användarens behov.

### 3.3 Länkar

GUI värdserver: <http://130.240.200.103:5000/>

Github projekt: <https://github.com/mebakid74/Human-Health-and-Activity-Laboratory-H2A1-UWB-navigering-av->

Github scrum tabel: <https://github.com/users/mebakid74/projects/3/views/4?visibleFields=%5B%22Title%22%2C27215106%2C27215107%2C35389246%2C%22Status%22%2C27481975%5D>

## Referenser

- [1] *Aktivitetslaboratoriet*. Dec. 2022. URL: <https://www.ltu.se/centres/eic/Aktivitetslaboratoriet>.
- [2] *Foodora testar självkörande robot för hemleveranser*. Dec. 2022. URL: [https://www.dagenshandel.se/article/view/781466/foodora\\_testar\\_sjalvkorande\\_robot\\_for\\_hemleveranser](https://www.dagenshandel.se/article/view/781466/foodora_testar_sjalvkorande_robot_for_hemleveranser).
- [3] *Pepper (robot)*. Nov. 2022. URL: [https://en.wikipedia.org/wiki/Pepper\\_\(robot\)](https://en.wikipedia.org/wiki/Pepper_(robot)).
- [4] *Robotdammsugare*. Dec. 2022. URL: <https://sv.wikipedia.org/wiki/Robotdammsugare>.
- [5] *Robotic vacuum cleaner*. Dec. 2022. URL: [https://en.wikipedia.org/wiki/Robotic\\_vacuum\\_cleaner](https://en.wikipedia.org/wiki/Robotic_vacuum_cleaner).
- [6] *Robotik*. Dec. 2022. URL: <https://sv.wikipedia.org/wiki/Robotik>.
- [7] *Shopping mall robots are boosting retail*. Dec. 2022. URL: <https://www.electronicsspecifier.com/industries/robotics/shopping-mall-robots-are-boosting-retail>.
- [8] *Ultra-wideband*. Dec. 2022. URL: <https://en.wikipedia.org/wiki/Ultra-wideband>.
- [9] *UNIMATE*. Dec. 2022. URL: <https://www.automate.org/a3-content/joseph-engelberger-unimate>.
- [10] *ZeroKey-API*. Mars 2023. URL: <https://api.zerokey.com/#section/Event-Categories/HARDWARE/>.
- [11] *ZeroKey-UWB*. Mars 2023. URL: <https://zerokey.com/quantum-rtls/>.