

Glassify

Transforming **smartphone UIs** into
OHMD experiences

Charisma Kausar

FYP Student at Synteraction Lab



Motivation



Motivation



How would you place a food delivery order
when you're busy cooking?

Motivation



How would you place a food delivery order
when you're busy cooking?

Motivation



Smartphones with
Voice Interaction



How would you place a food delivery order
when you're busy cooking?

Motivation



How would you place a food delivery order
when you're busy cooking?

New Opportunity



Wide array of
smartphone apps

+

Multitasking ability
of OHMDS



New Opportunity



Wide array of
smartphone apps

+

Multitasking ability
of OHMDS



Requires a smartphone-to-OHMD
UI transformation pipeline

Agenda

- Motivation
- Research Problem
- Related Work
- Glassify Prelude
- Methodology
- Evaluation
- Discussion
- Future Work
- Demo
- Conclusion

Research Question 1

**How can smartphone GUIs
be transformed into
multitasking-friendly OHMD interfaces?**

Research Question 1

**How can smartphone GUIs
be transformed into
multitasking-friendly OHMD interfaces?**

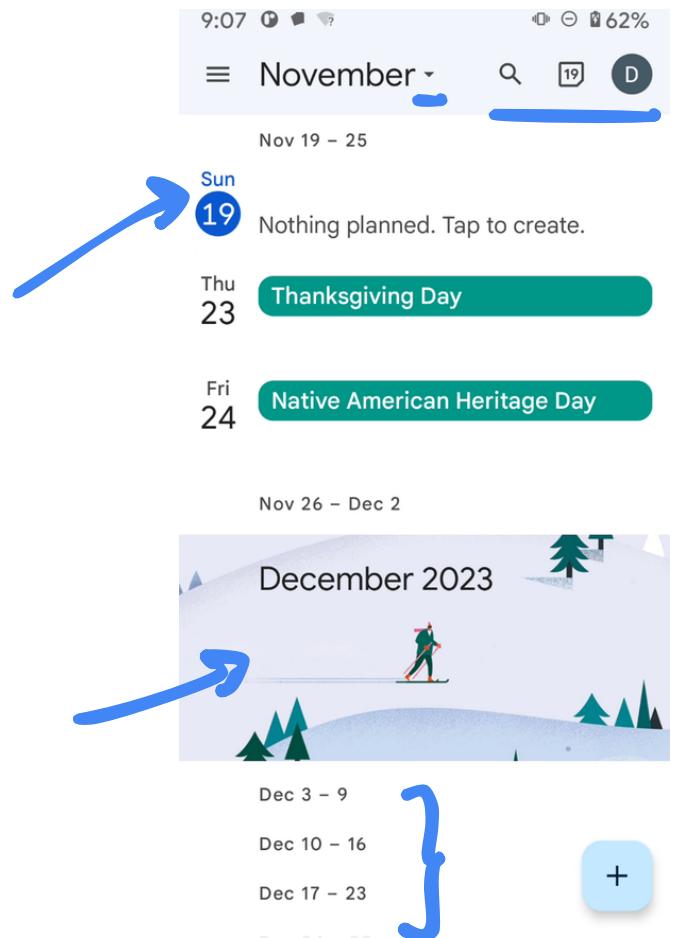
Research Question 2

**How can touch interactions
be transformed into
OHMD-friendly voice-based interactions?**

RQ1. Interface design challenges

RQ1. Interface design challenges

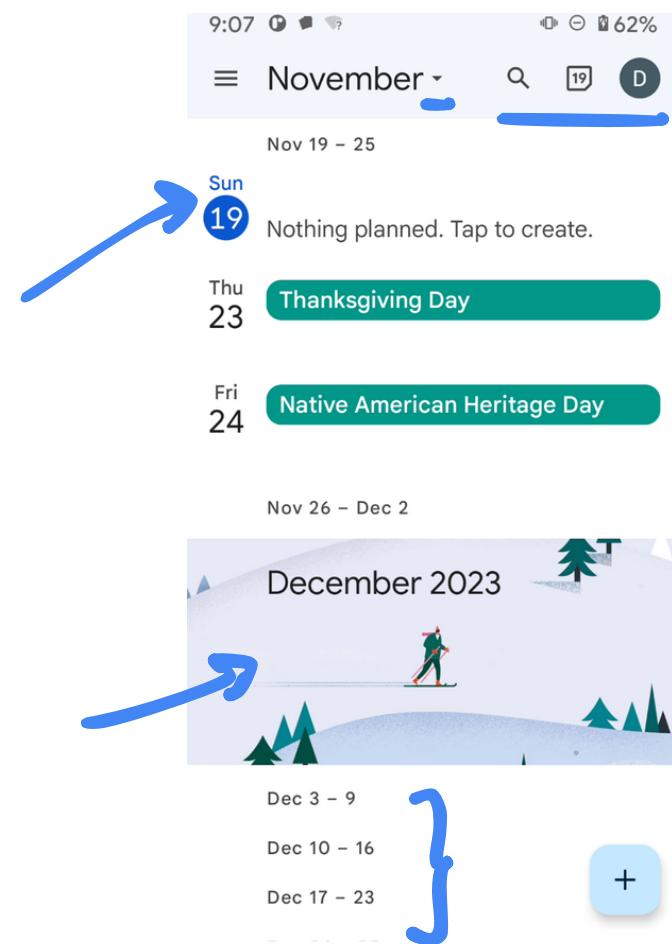
1. Complexity reduction



Need to simplify
complex
smartphone UIs

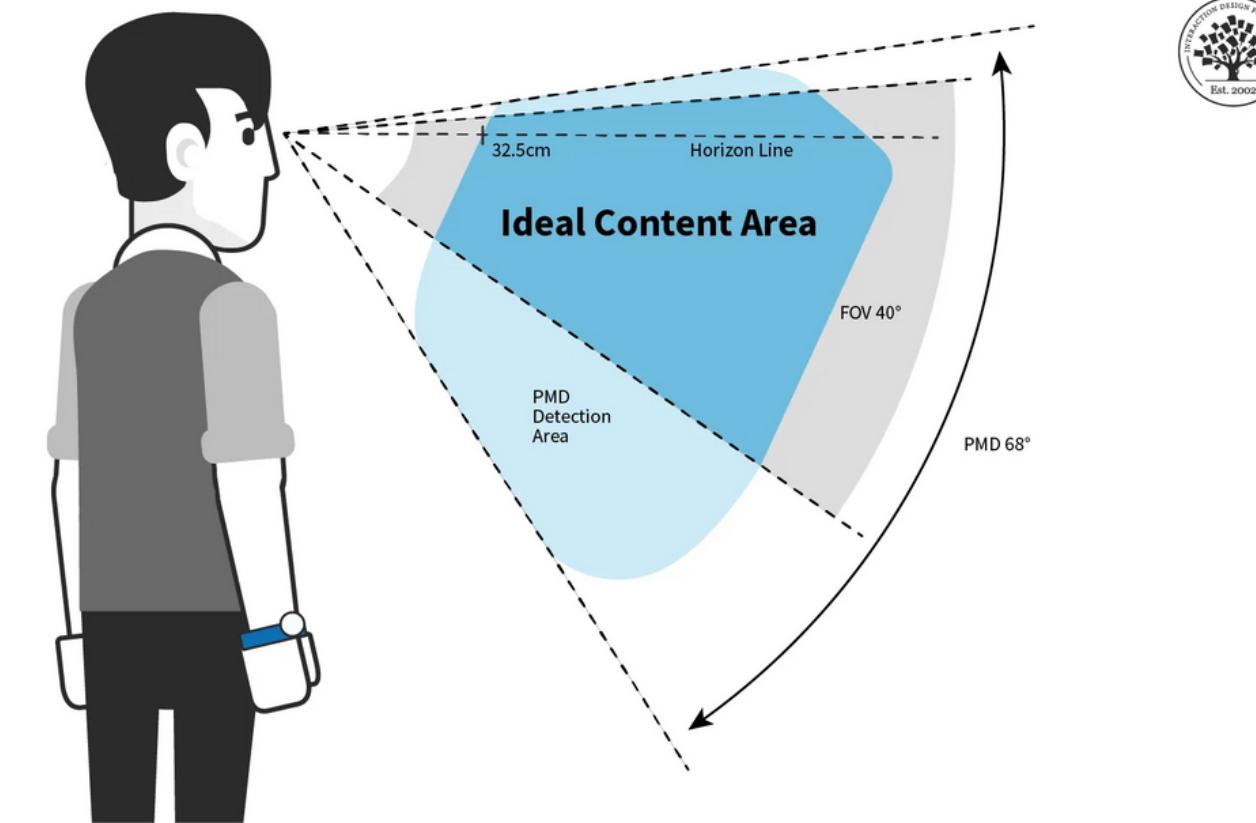
RQ1. Interface design challenges

1. Complexity reduction



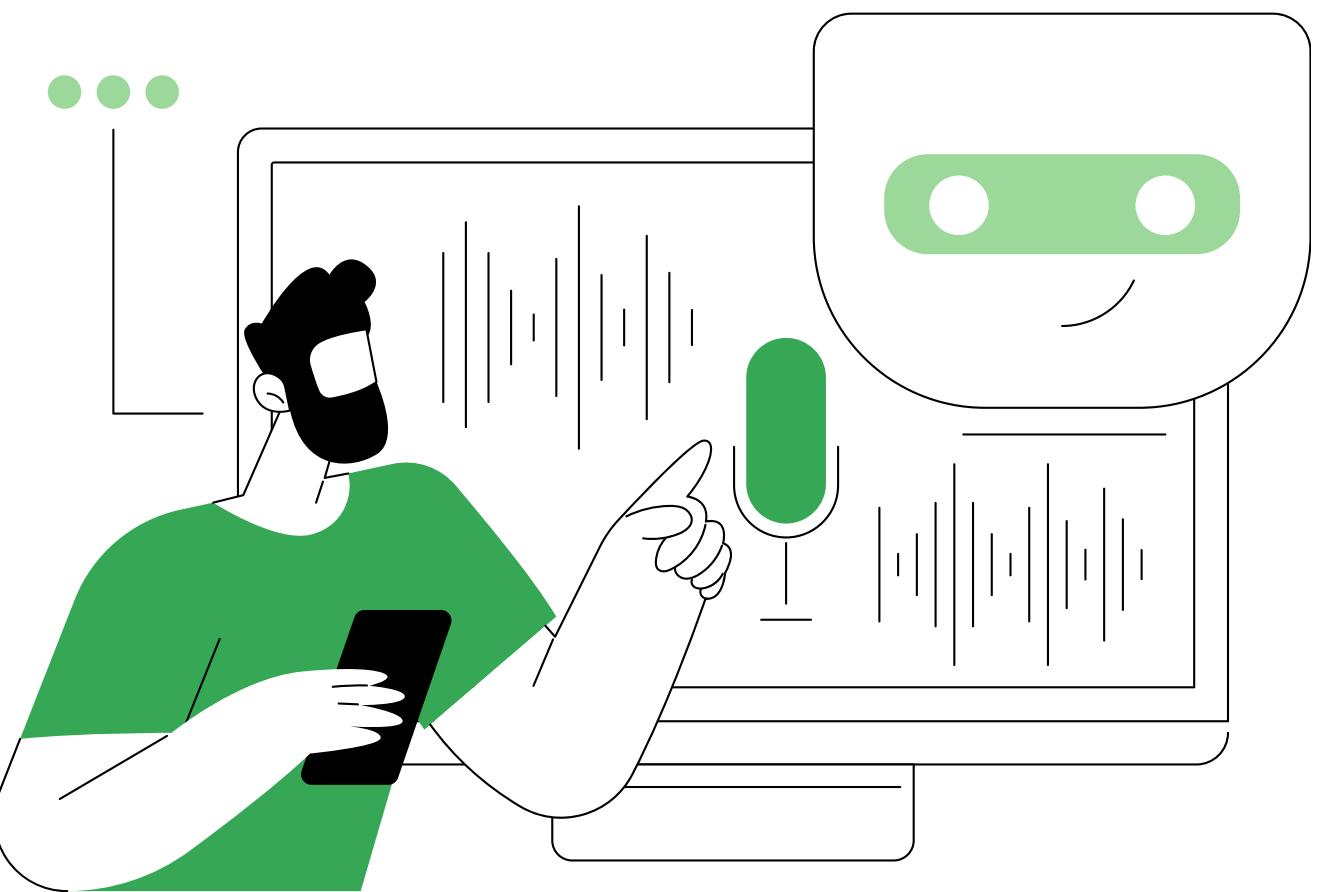
Need to simplify
complex
smartphone UIs

2. Field of view utilization



Utilize FoV while minimising distractions

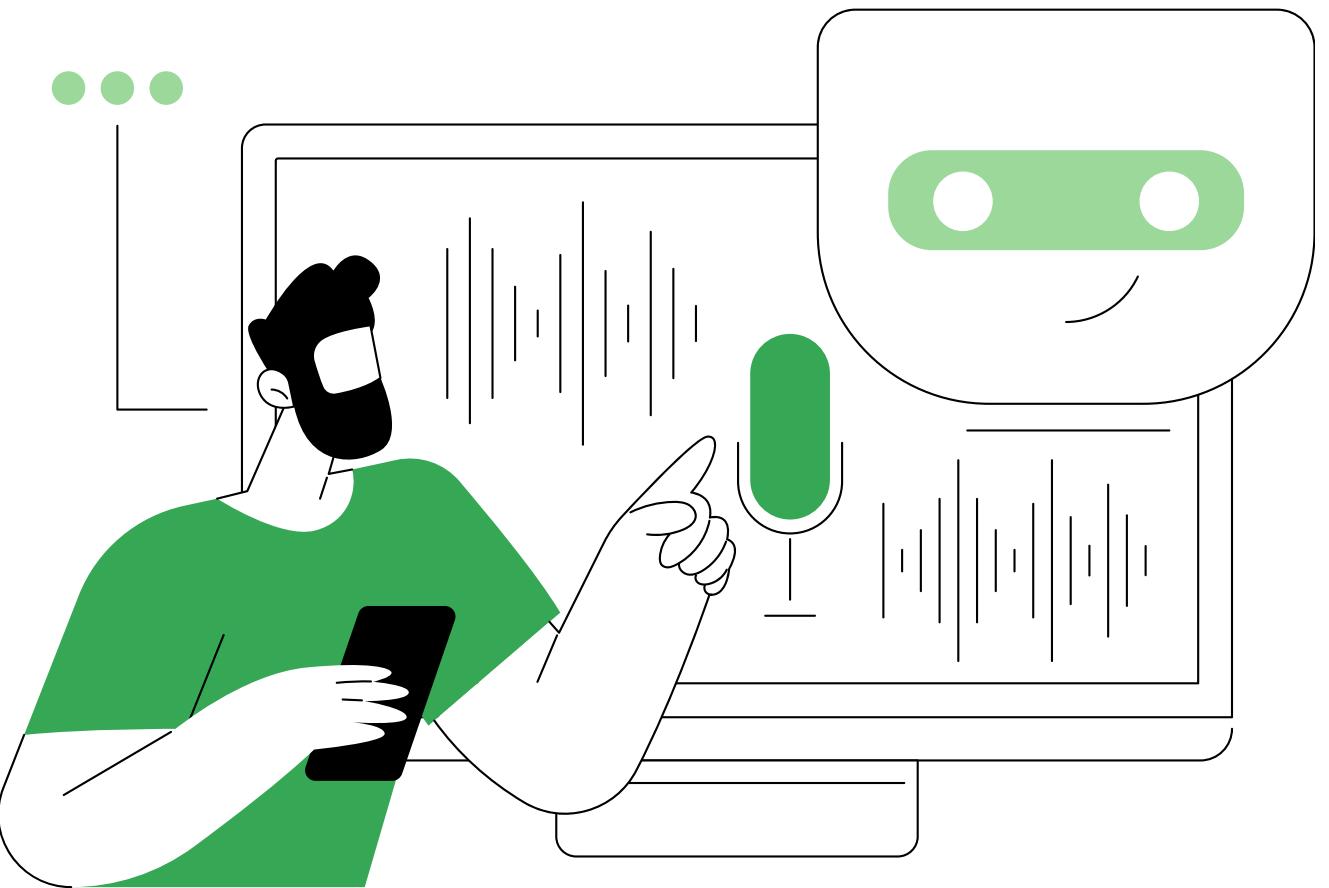
RQ2. Interaction design challenges



RQ2. Interaction design challenges

1. Voice command design

Craft intuitive commands



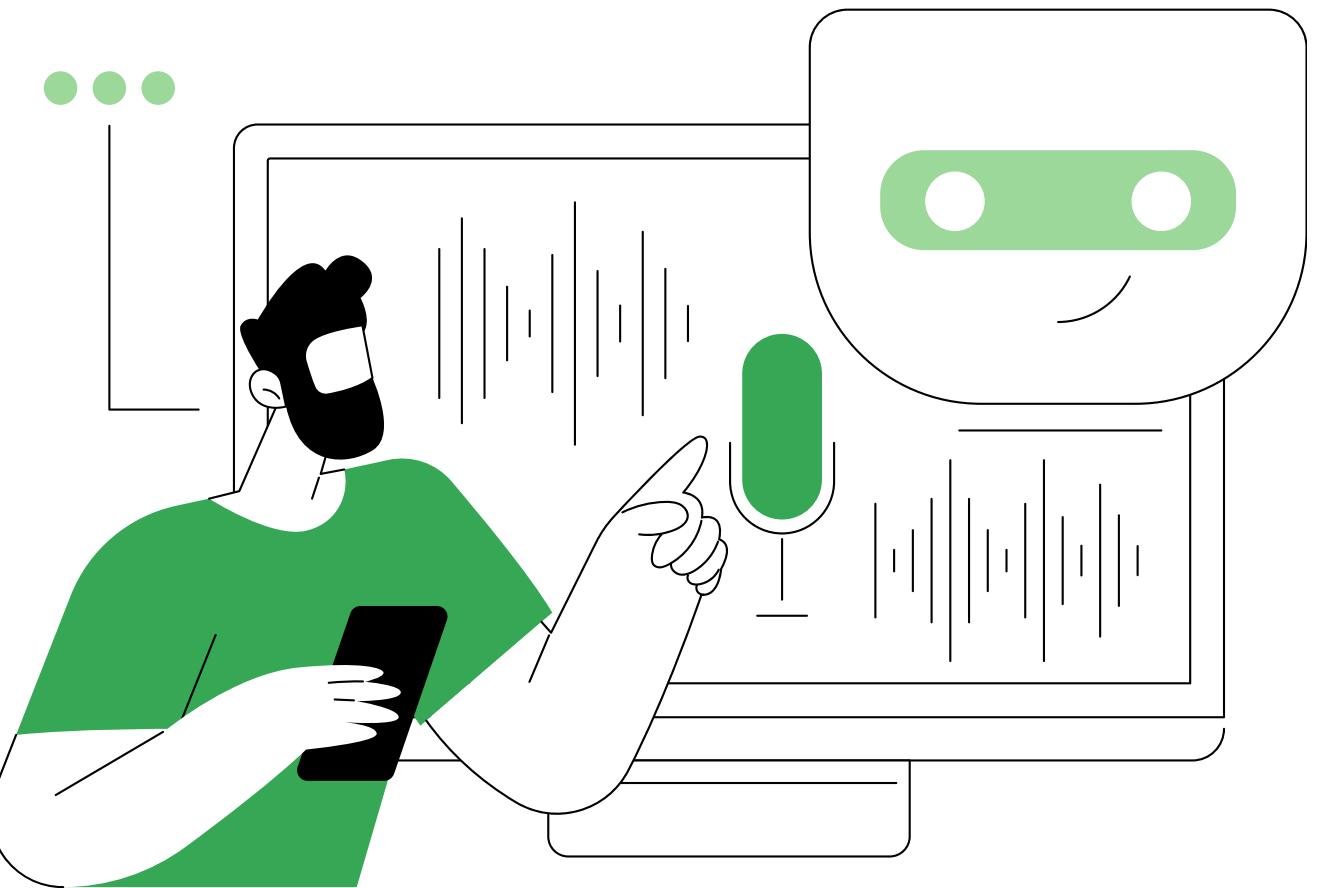
RQ2. Interaction design challenges

1. Voice command design

Craft intuitive commands

2. Feedback and error handling

Minimise errors as voice interaction is slower



RQ2. Interaction design challenges

1. Voice command design

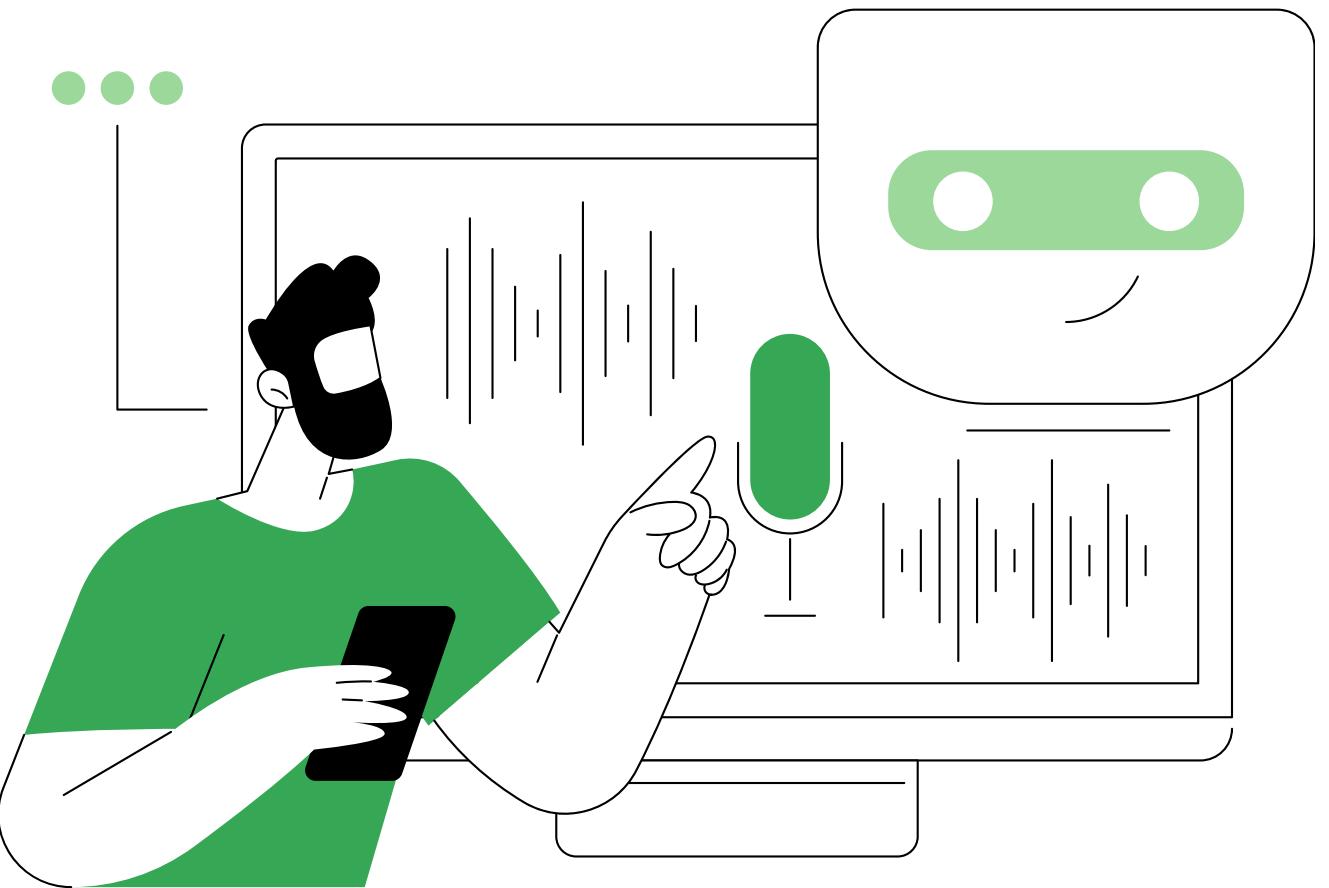
Craft intuitive commands

2. Feedback and error handling

Minimise errors as voice interaction is slower

3. Interface adaptation

Adapt touch-optimised UIs to make them more voice-friendly



Agenda

Motivation

Research Problem

Related Work

Glassify Prelude

Methodology

Evaluation

Discussion

Future Work

Demo

Conclusion

Related Work



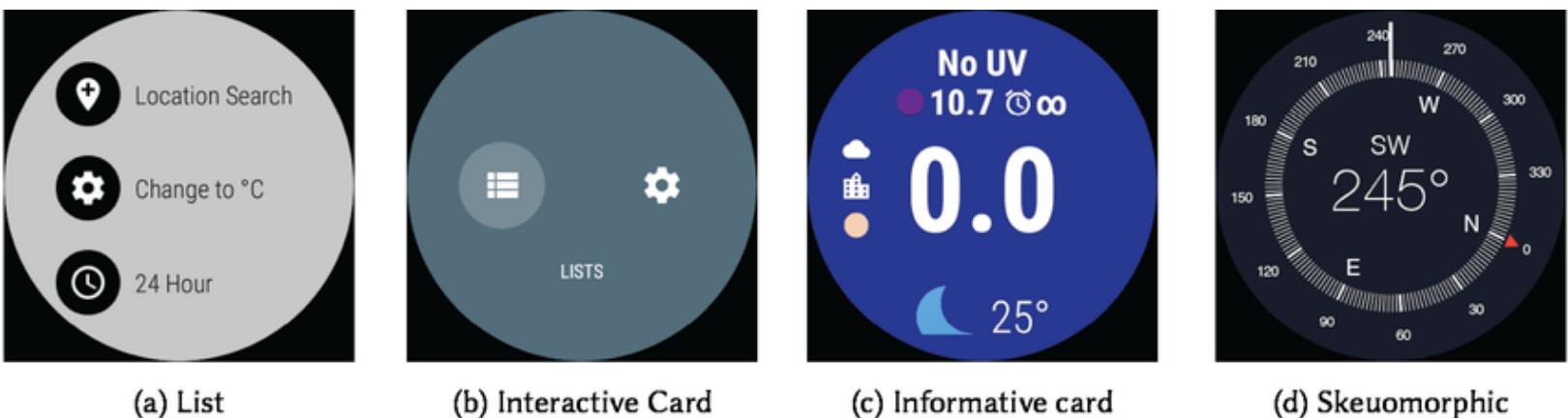
- 1. Focus on single app type**
eg. GlassMessaging for messages

Related Work



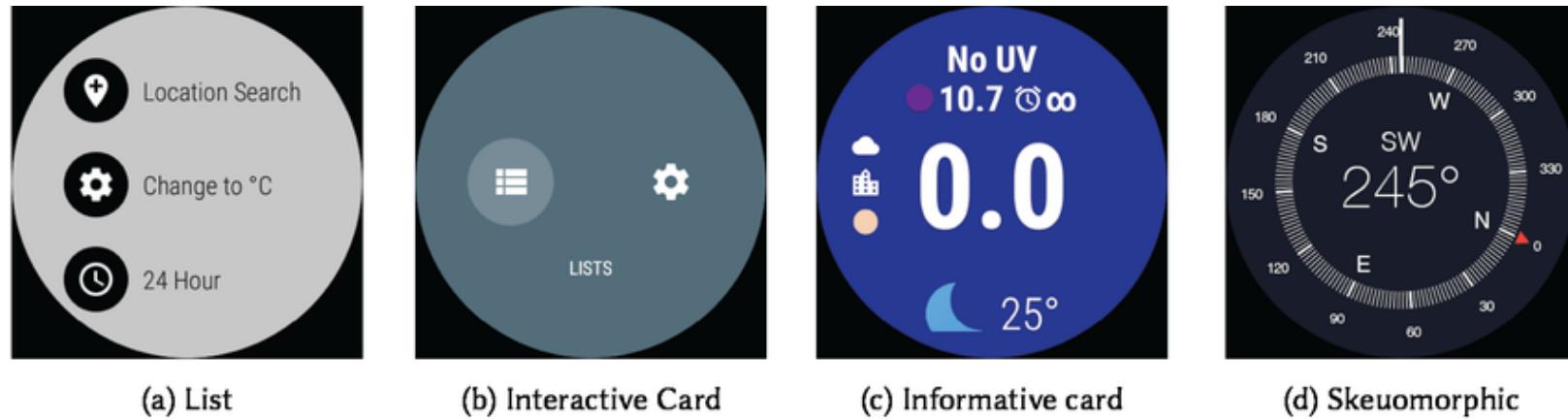
1. Focus on single app type

eg. GlassMessaging for messages



2. Adaptive UI for other paradigms

Related Work



1. Focus on single app type
eg. GlassMessaging for messages

2. Adaptive UI for other paradigms
eg. Smartphone to smartwatch

Challenges of directly mapping smartphone-based UIs into OHMD UIs are untouched.

Glassify

**Guidelines for transforming smartphone UIs into OHMD interfaces
for multitasking scenarios with voice control.**

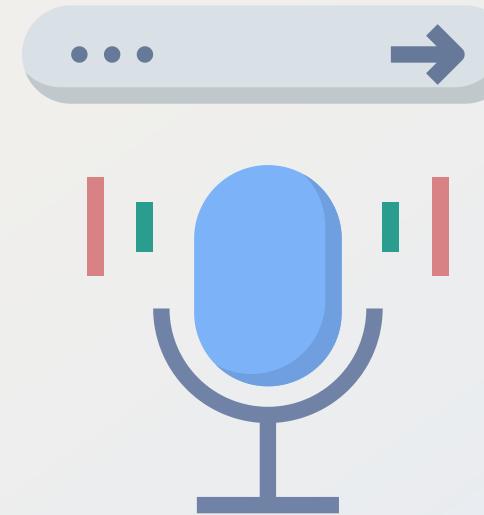


PRELUDE



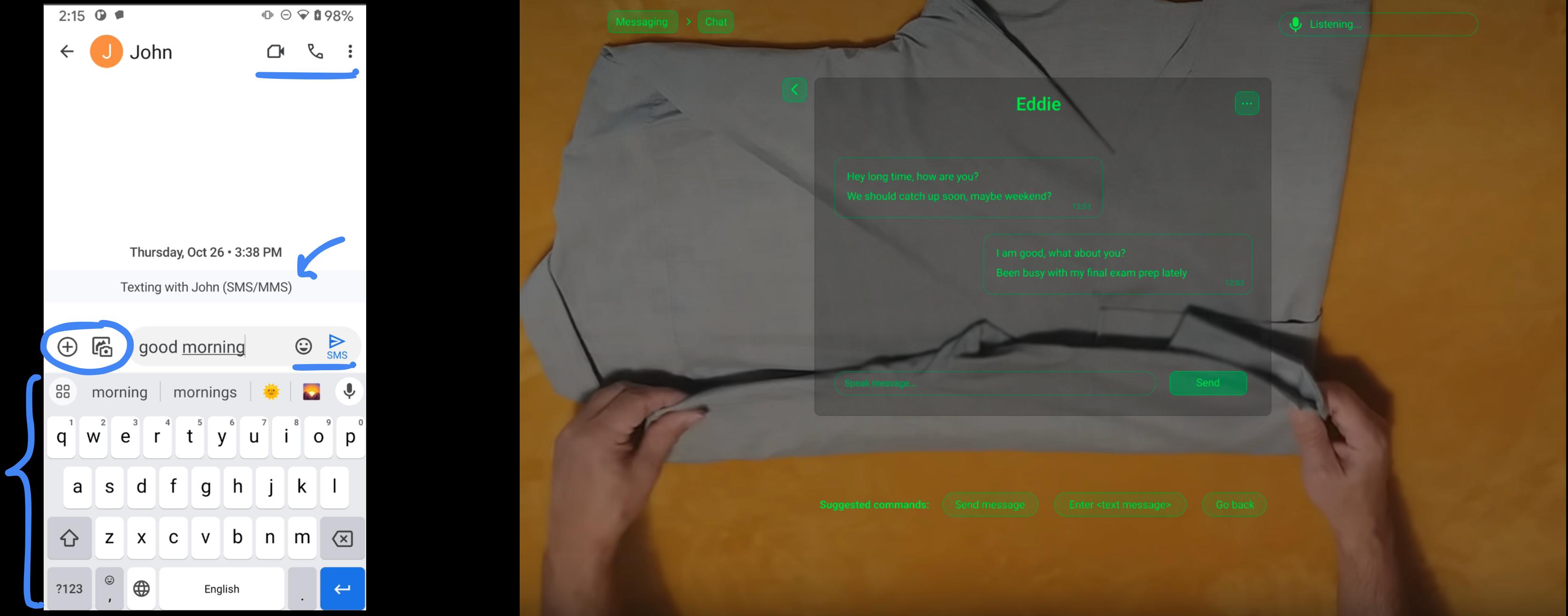
Interface design

+



Interaction design

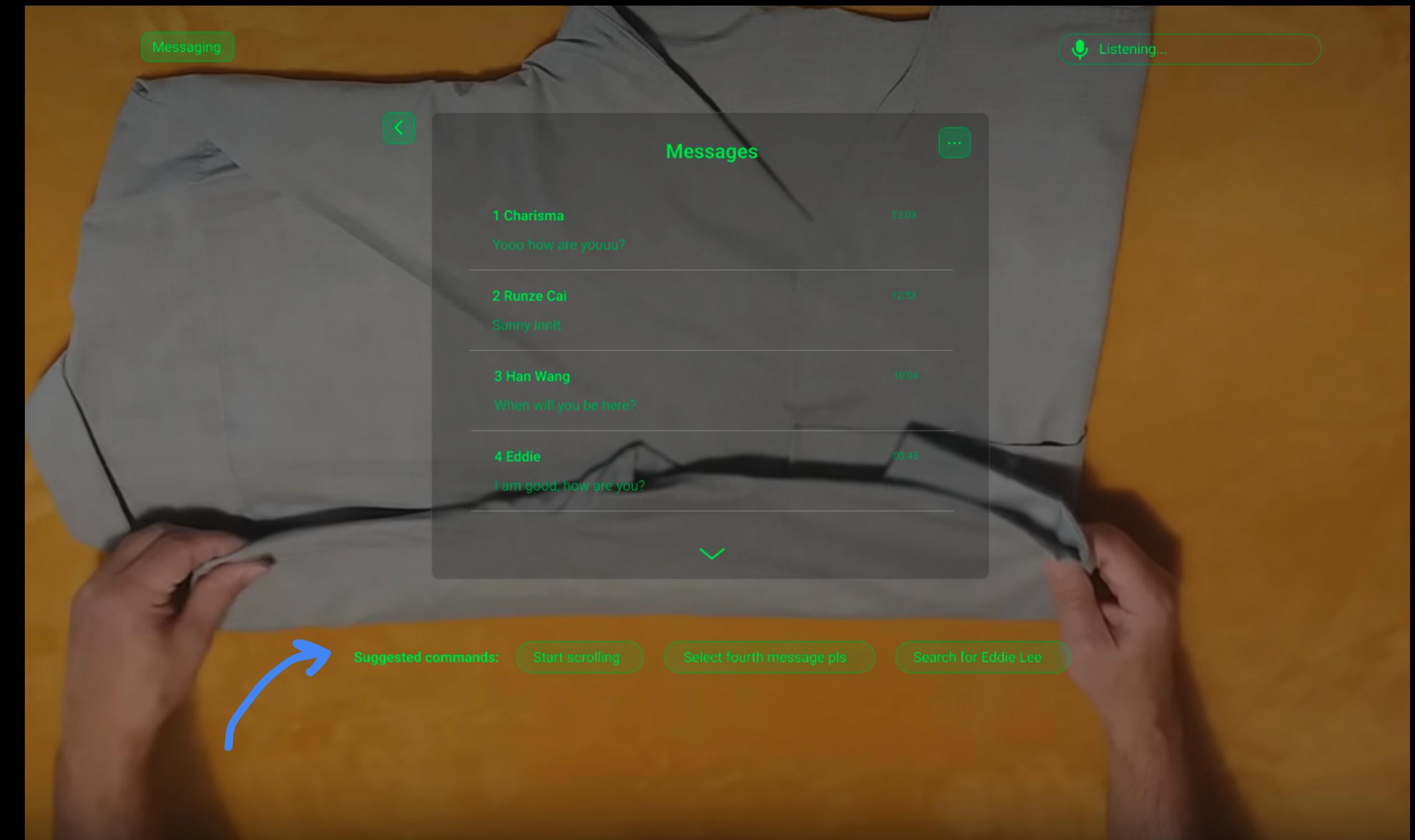
Interface design



Interaction design

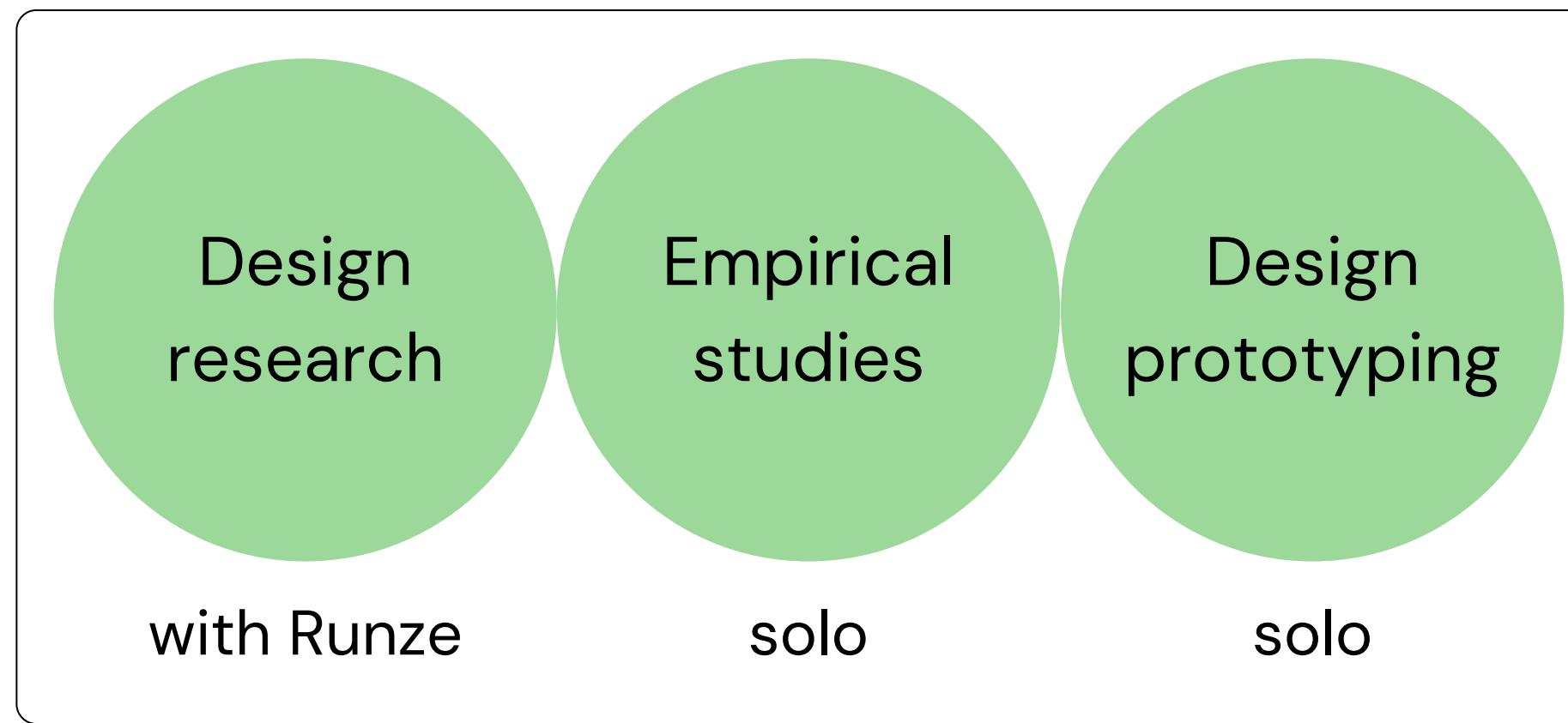
“Start scrolling”

“Select message 4”

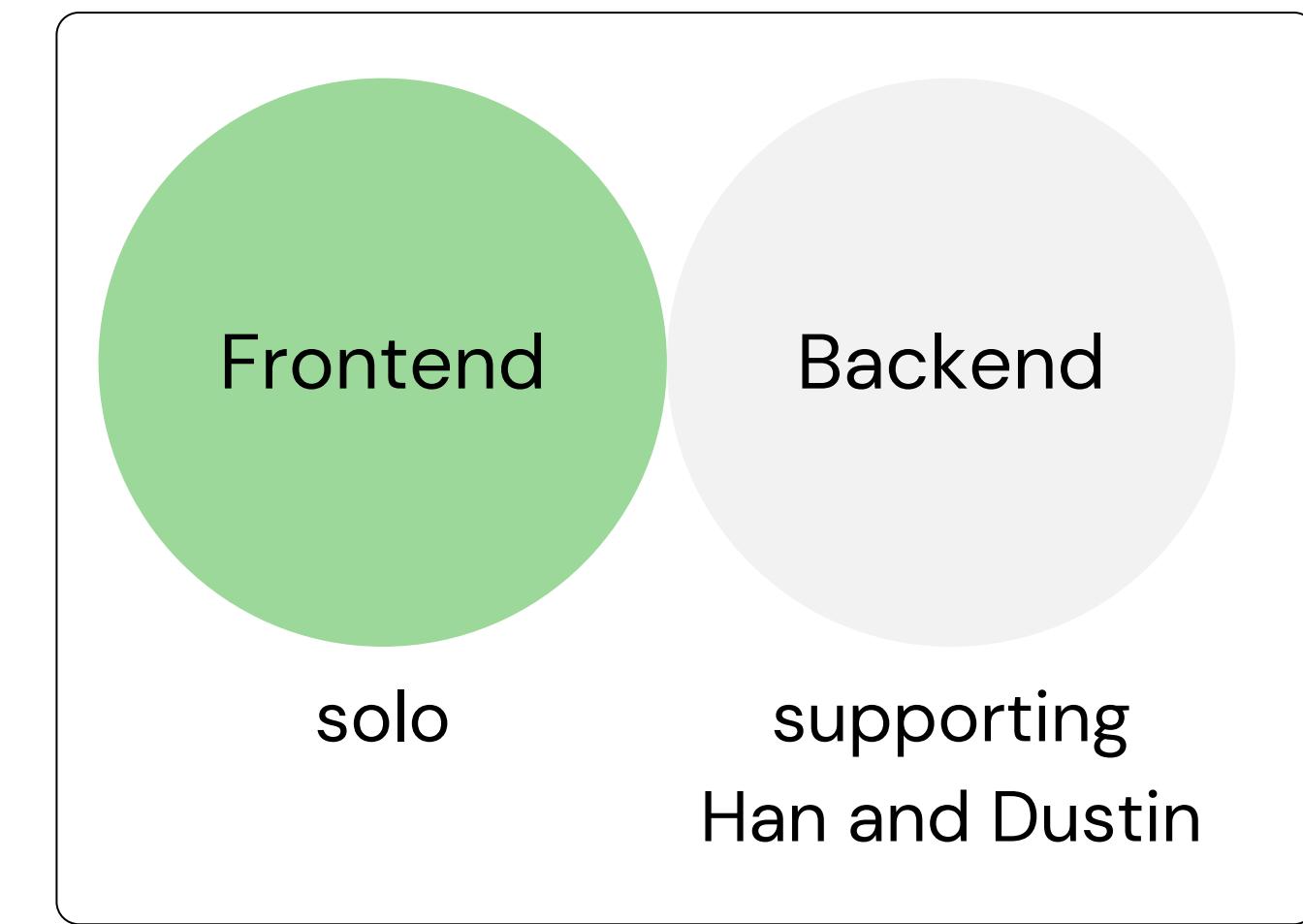


Contributions

Collaboration between **NUS** (Runze – PhD, and me)
and **Monash University** (Han – PhD, and Dustin – Graduate TA)



Design

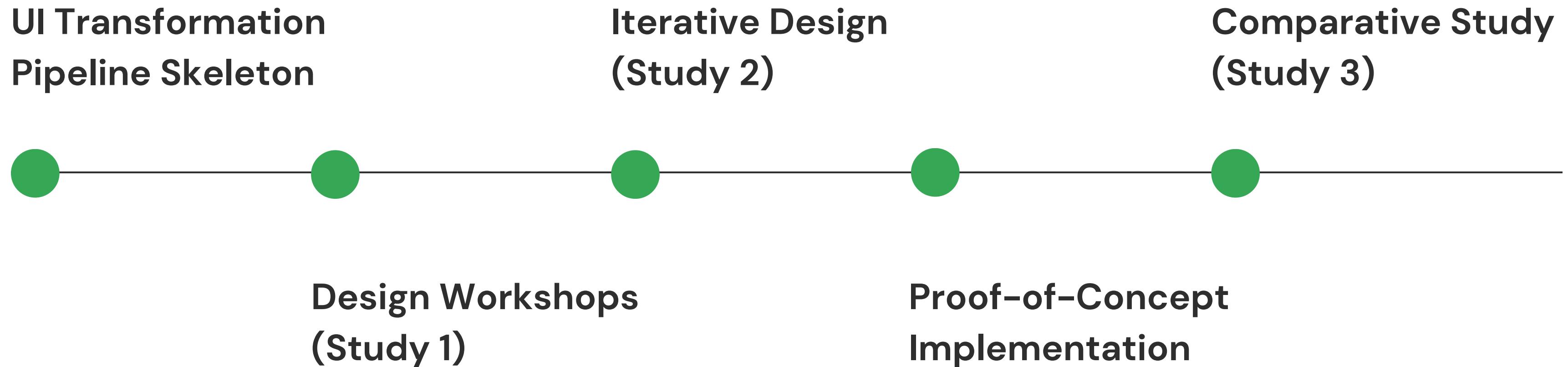


Implementation

Agenda

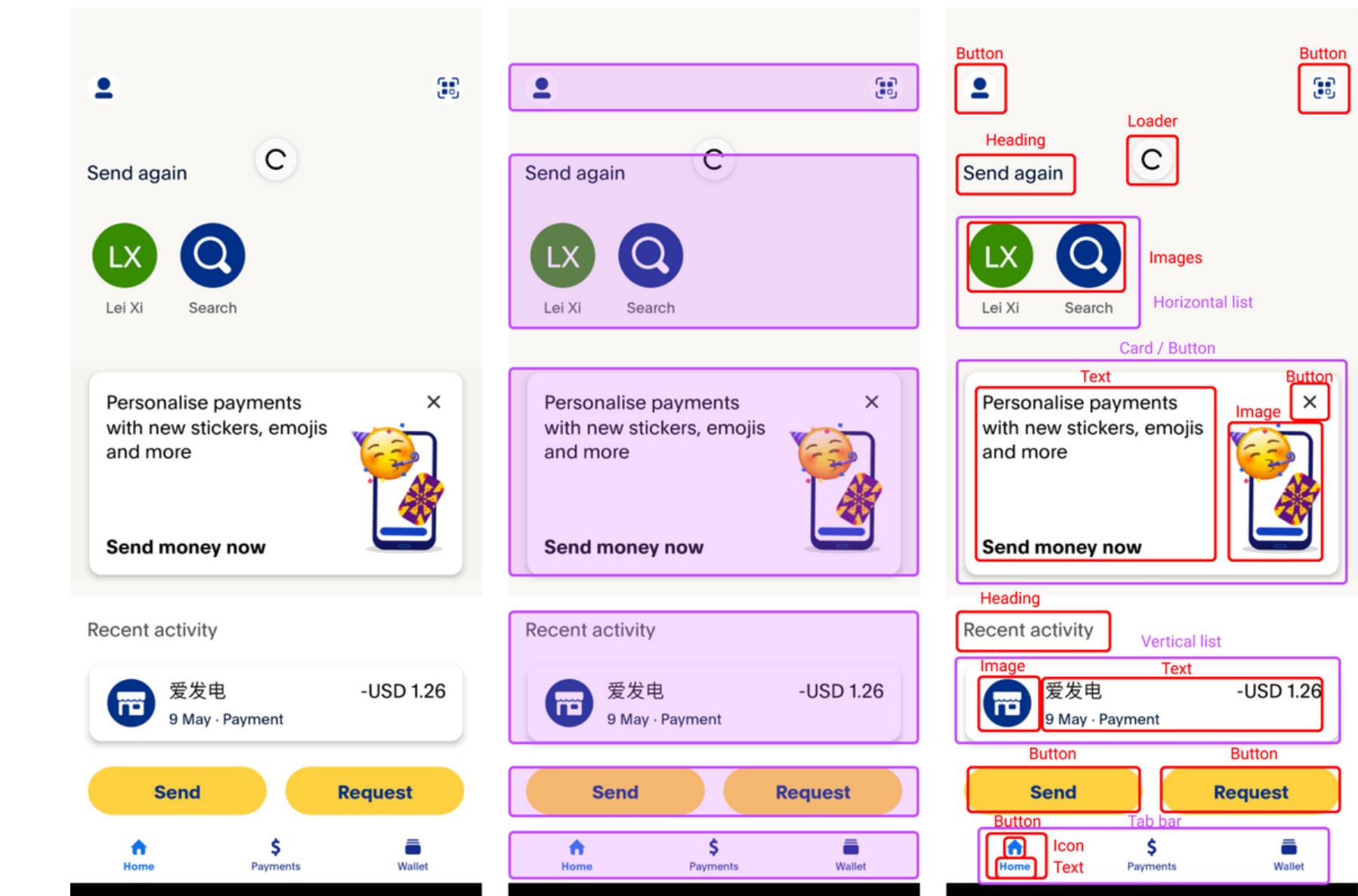
- Motivation
- Research Problem
- Related Work
- Glassify Prelude
- Methodology**
- Evaluation
- Discussion
- Future Work
- Demo
- Conclusion

Methodology Overview



UI Transformation Pipeline Skeleton

1. Decomposition/Analysis



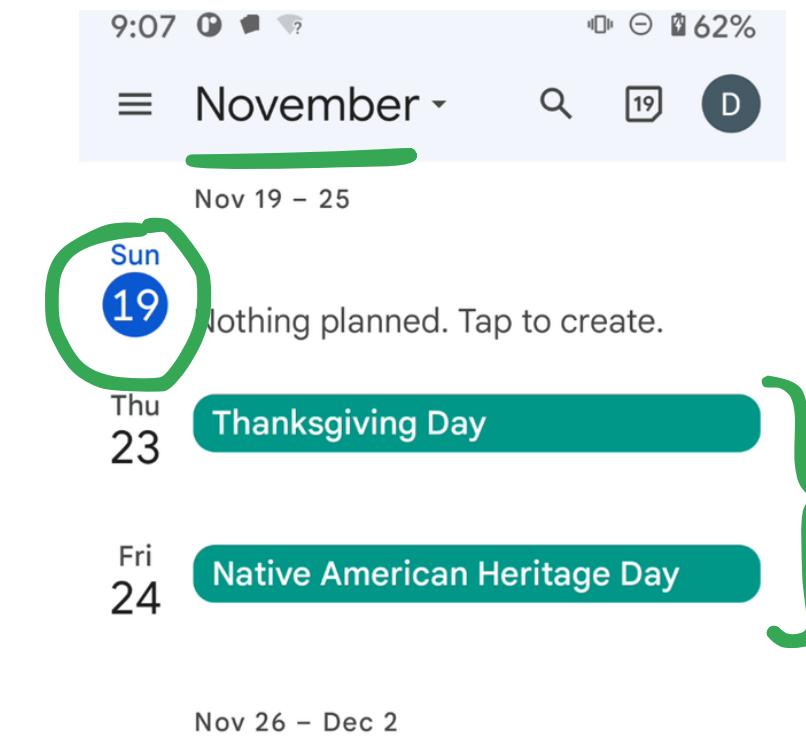
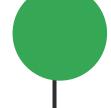
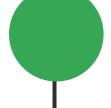
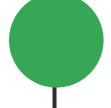
UI Transformation Pipeline Skeleton



1. Decomposition/Analysis

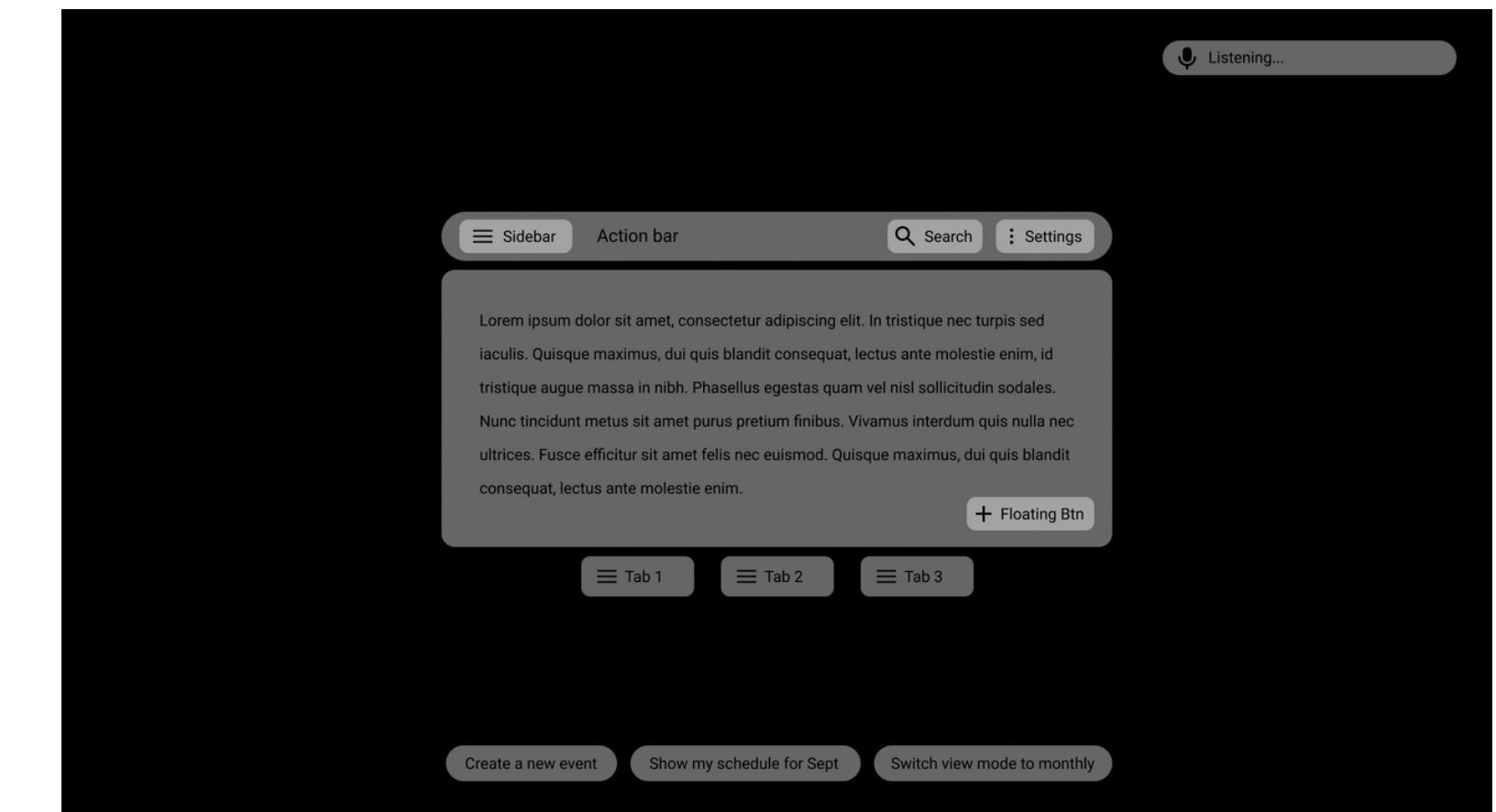


2. Filter/Prioritisation



UI Transformation Pipeline Skeleton

1. Decomposition/Analysis
2. Filter/Prioritisation
3. Simple layout transformation



UI Transformation Pipeline Skeleton

1. Decomposition/Analysis
2. Filter/Prioritisation
3. Simple layout transformation
4. Voice-friendly enhancements

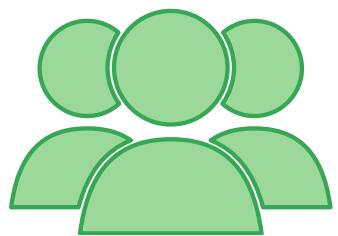


UI Transformation Pipeline Skeleton

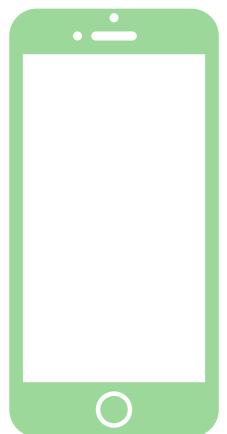
- 1. Decomposition/Analysis
- 2. Filter/Prioritisation
- 3. Simple layout transformation
- 4. Voice-friendly enhancements
- 5. Special case handling



Design Workshops

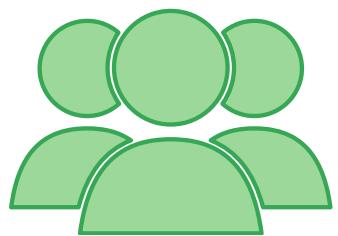


12 experienced AR researchers

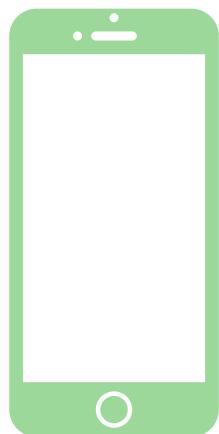


Transform a selected
smartphone app into OHMD
interfaces and discuss
considerations

Design Workshops



12 experienced AR researchers

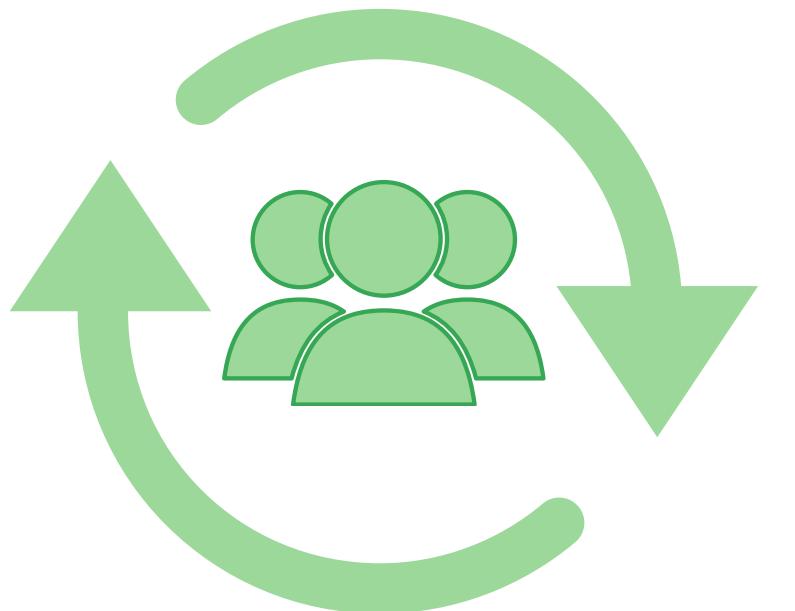


Transform a selected smartphone app into OHMD interfaces and discuss considerations

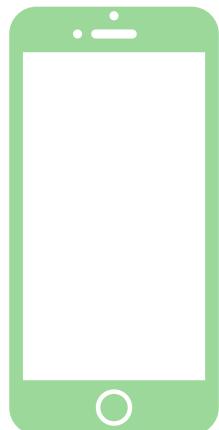
Findings

- Prioritisation framework
- Voice-friendly shortcuts
- Special voice commands to speed up interaction
- Error handling techniques

Iterative Design

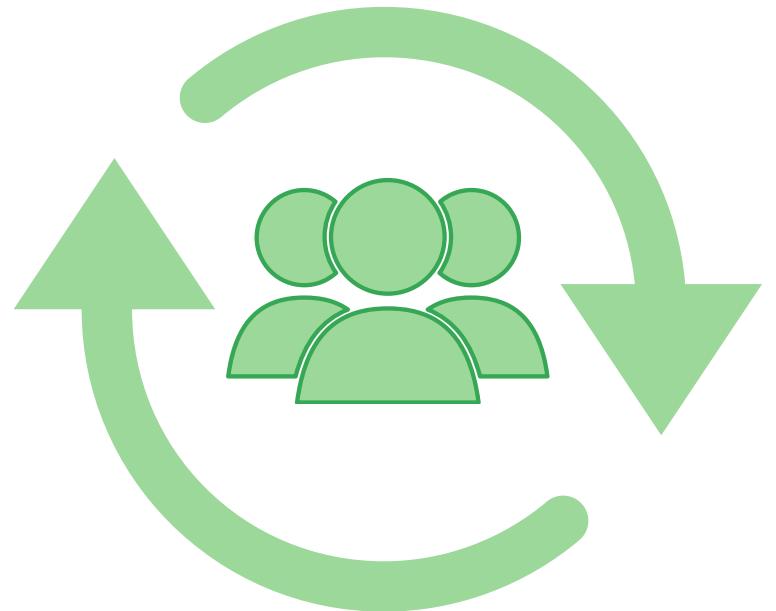


4 users pilot + 3 Iterations

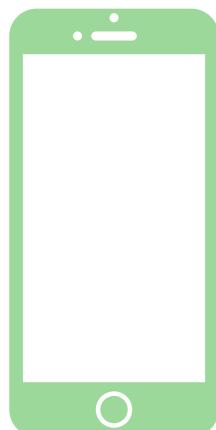


Converted 5 smartphone apps
into OHMD interfaces in Figma

Iterative Design



4 users pilot + 3 Iterations

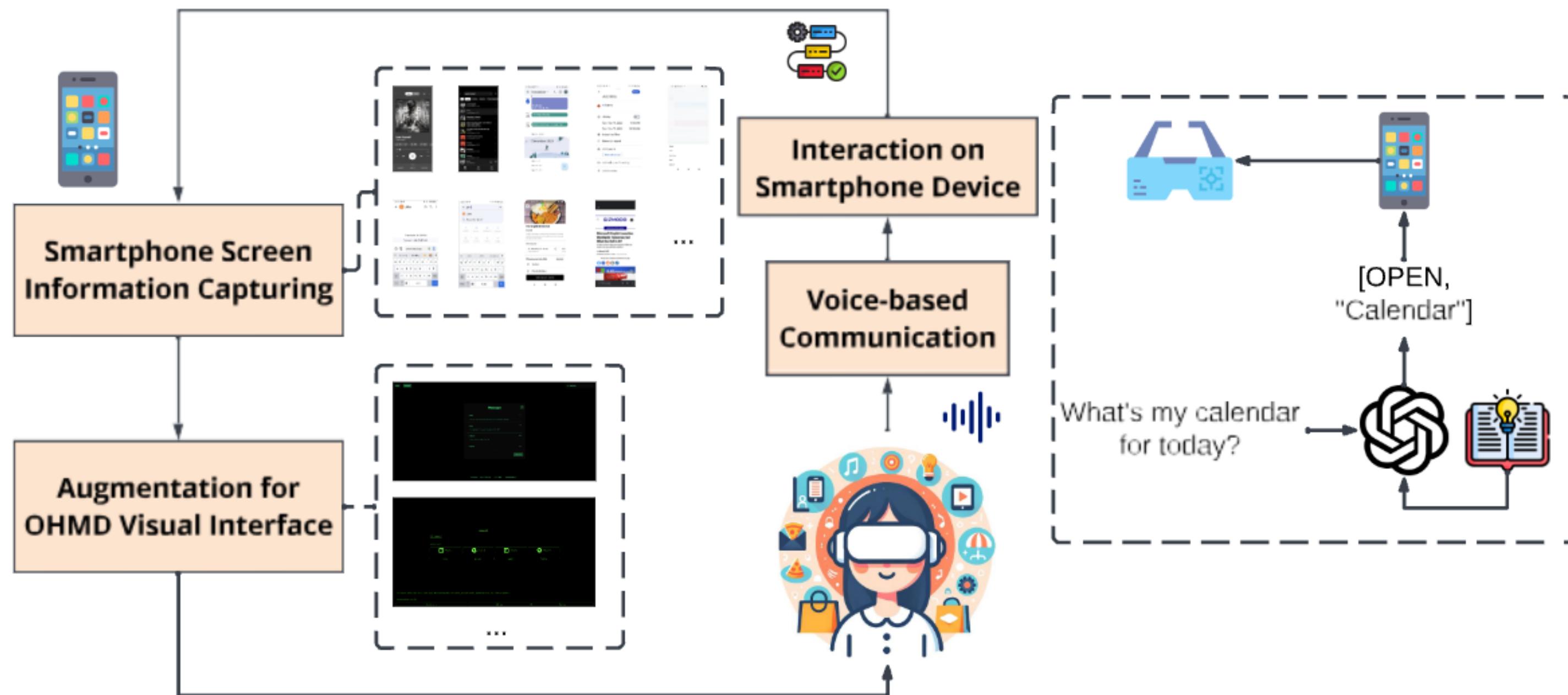


Converted 5 smartphone apps
into OHMD interfaces in Figma

Findings

- Creating a clear visual hierarchy in OHMD design
- Using a suitable list layout based on context

Implementation



Hardware Specifications



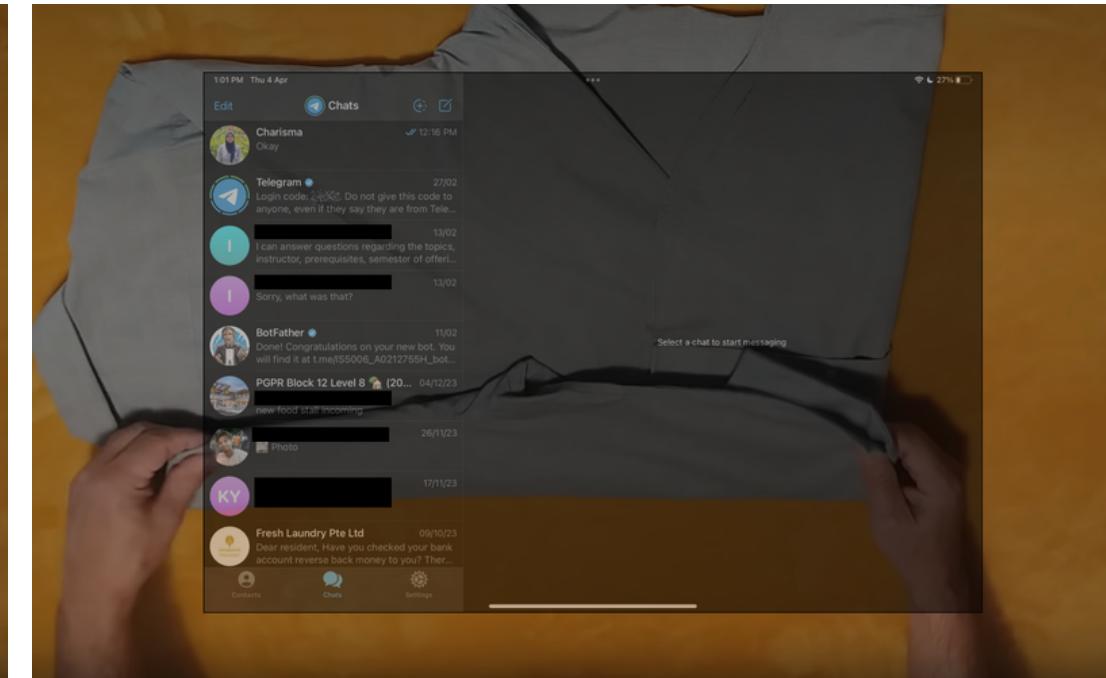
Agenda

- Motivation
- Research Problem
- Related Work
- Glassify Prelude
- Methodology
- Evaluation
- Discussion
- Future Work
- Demo
- Conclusion

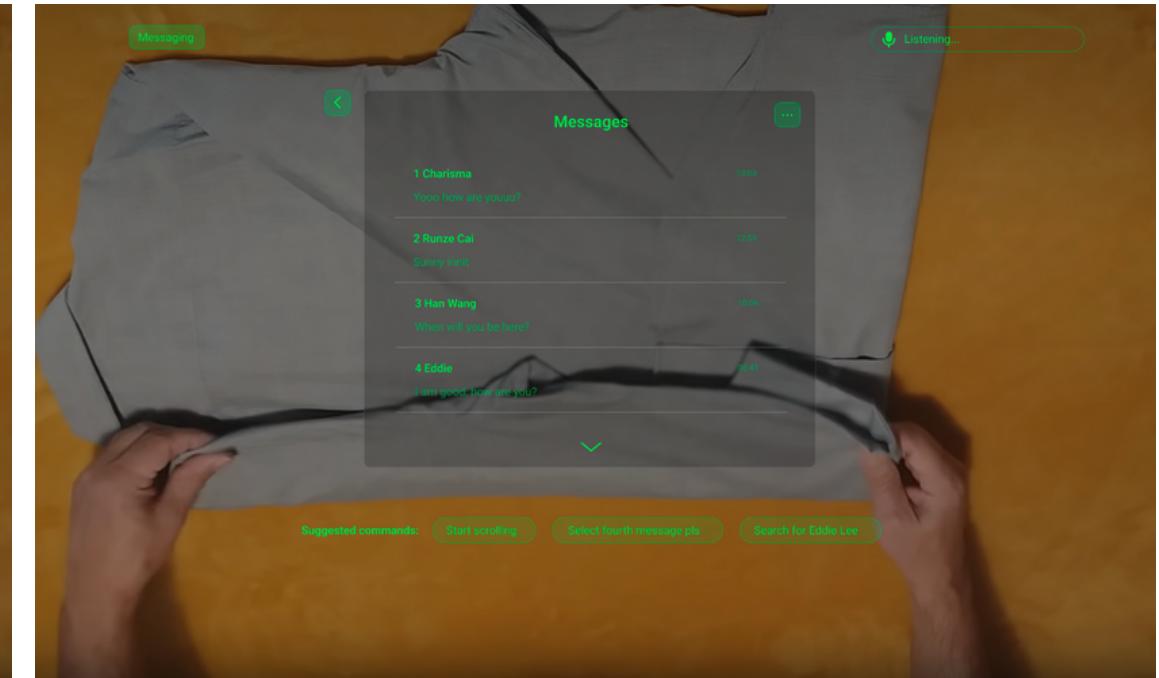
Comparative Study



Smartphone



VisionCast

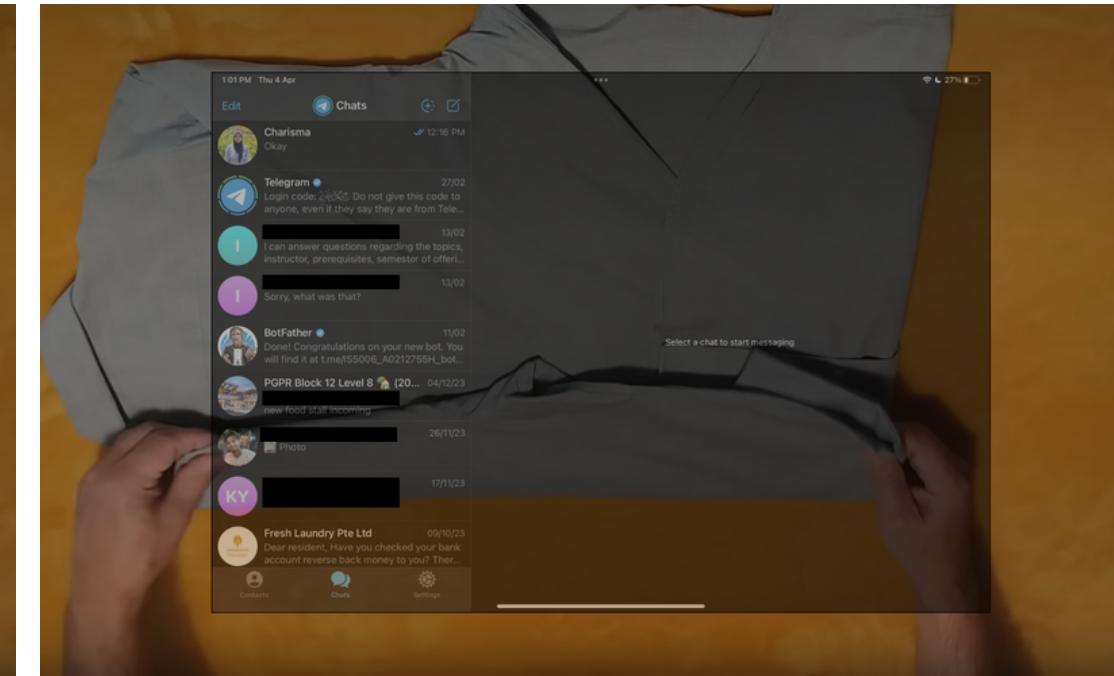


Glassify

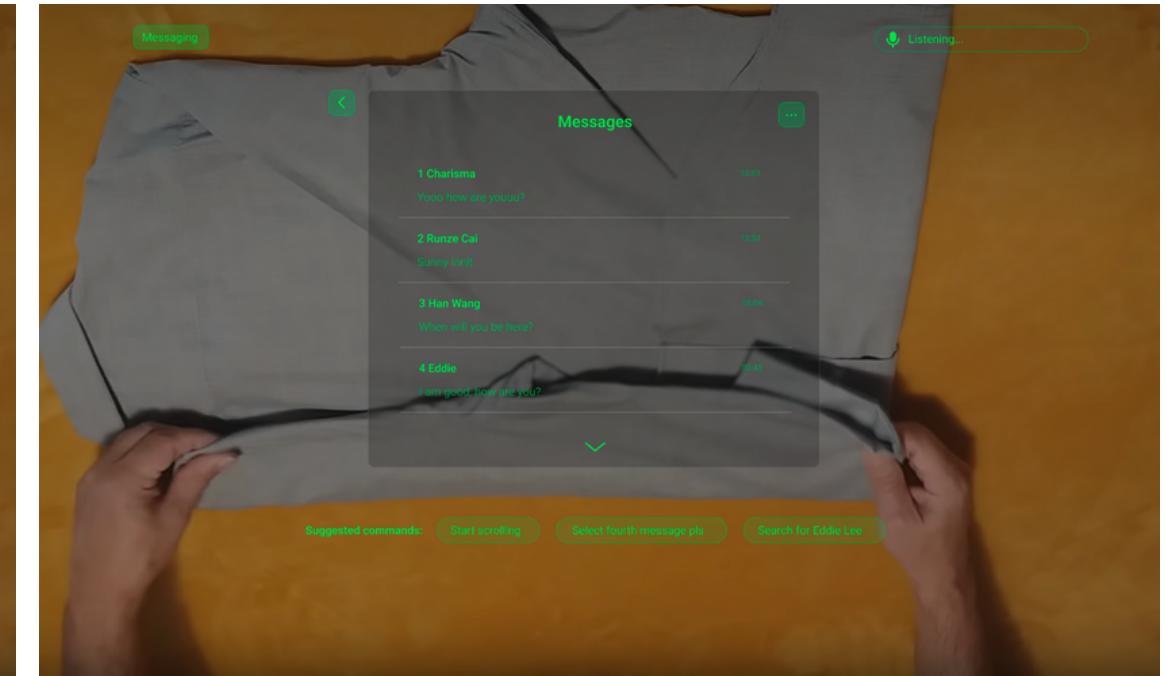
Comparative Study



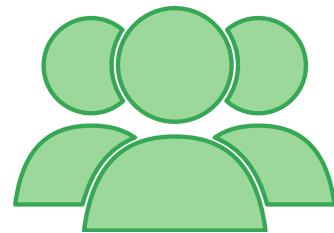
Smartphone



VisionCast

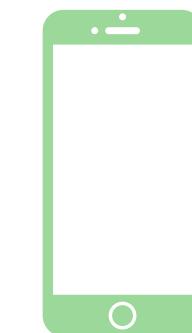


Glassify



6 participants

X

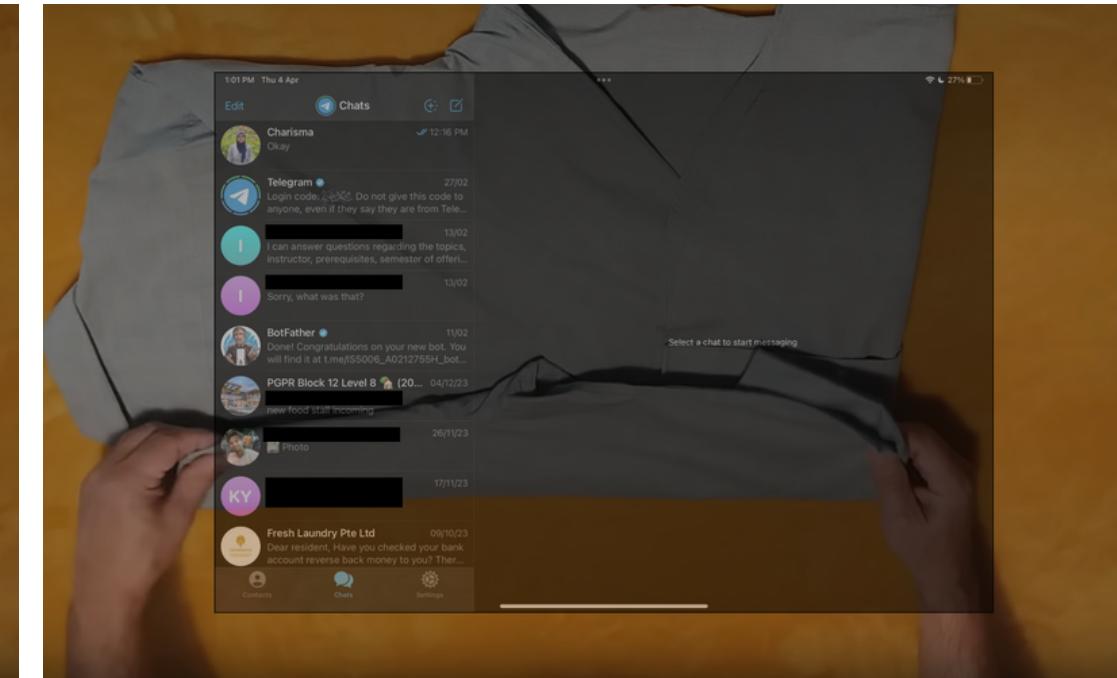


3 systems x 3 apps x 3 physical tasks = 27

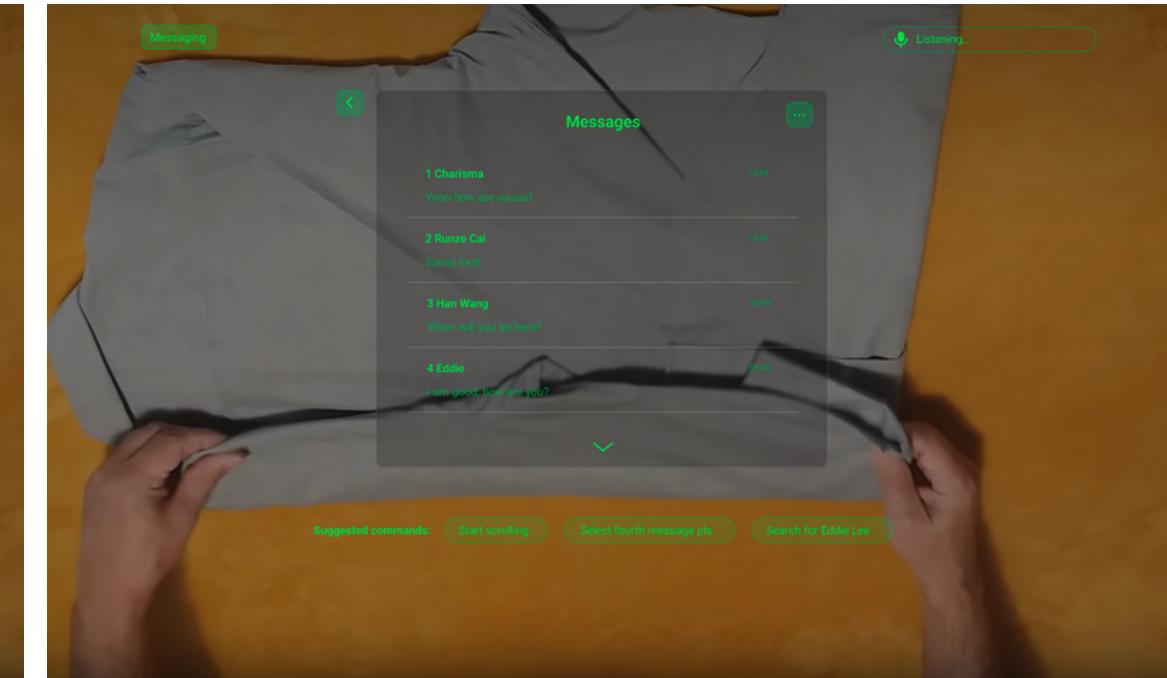
Comparative Study



Smartphone



VisionCast



Glassify



Distraction
Lowest



Usability
Highest



Workload
Lowest



Naturalness
Comparable

Comparative Study



Distraction
Lowest



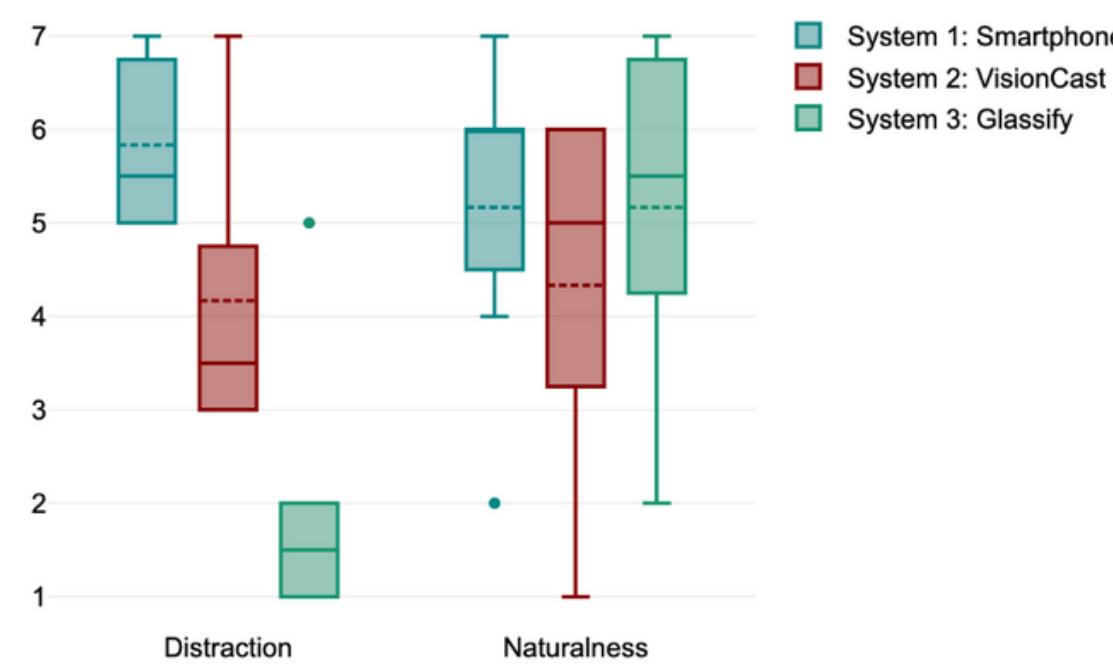
Usability
Highest



Workload
Lowest



Naturalness
Comparable



System	System Usability Scale (SUS) score
Smartphone	70.42
VisionCast	63.75
Glassify	74.17

System	Mental Workload (RAW-TLX) score
Smartphone	42.92
VisionCast	47.08
Glassify	30.42

Agenda

- Motivation
- Research Problem
- Related Work
- Glassify Prelude
- Methodology
- Evaluation
- Discussion
- Future Work
- Demo
- Conclusion

Discussion

Interface design

- Prioritisation framework
- Layout transformation
- Visual hierarchy

Interaction design

- Voice-friendly shortcuts
- Special voice commands
- Error handling techniques

Discussion

Interface design

- **Prioritisation framework**
- Layout transformation
- Visual hierarchy

1. Tasks analysis

Determine primary and secondary tasks of a screen

2. UI Element analysis

Determine whether each element is a Critical/Helpful/Non-essential element for the Primary/Secondary task types

Interaction design

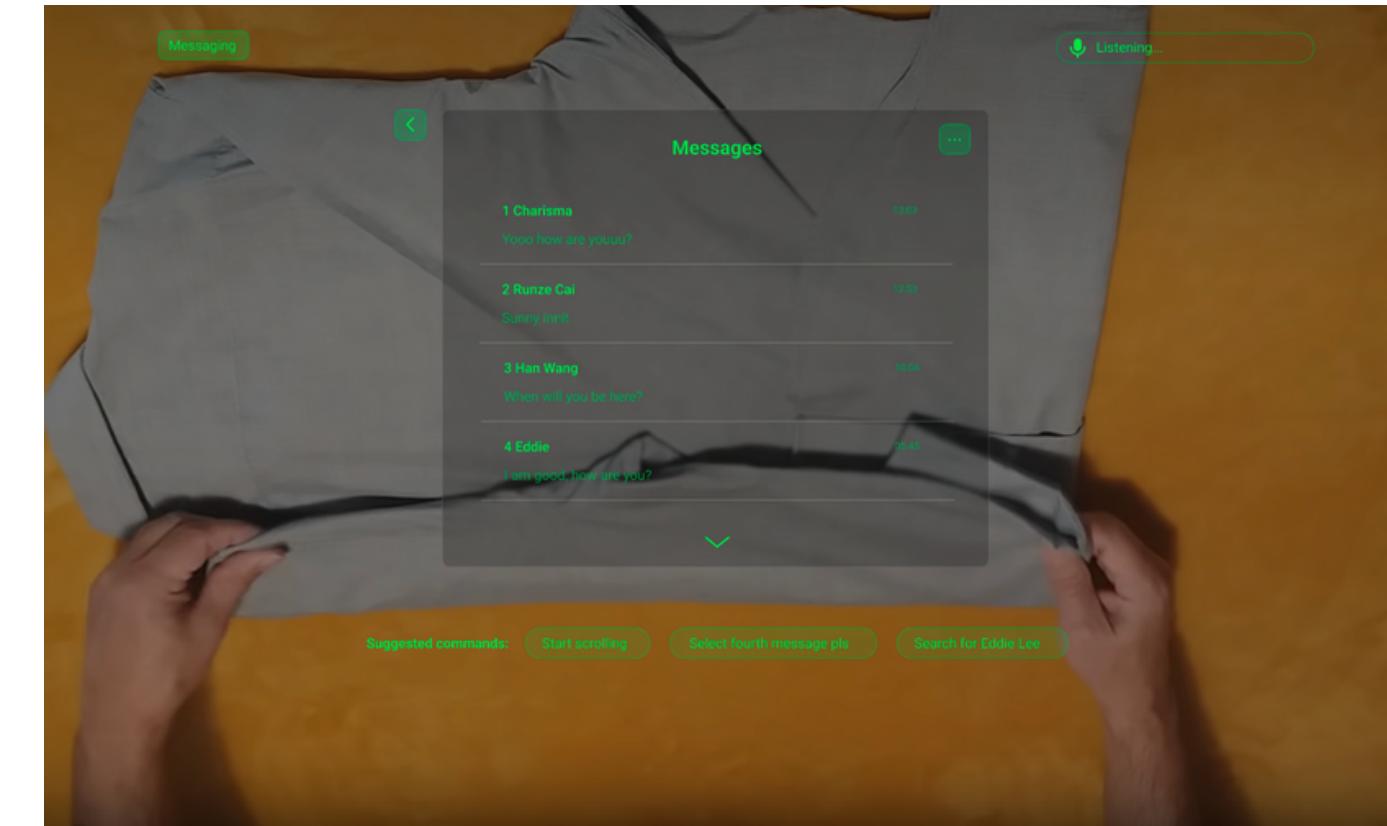
- Voice-friendly shortcuts
- Special voice commands
- Error handling techniques

"Glassify was much less distracting as the UI was simplified and less cluttered, which also made it easier to use and learn (P5)."

Discussion

Interface design

- Prioritisation framework
- **Layout transformation**
- Visual hierarchy



Interaction design

- Voice-friendly shortcuts
- Special voice commands
- Error handling techniques

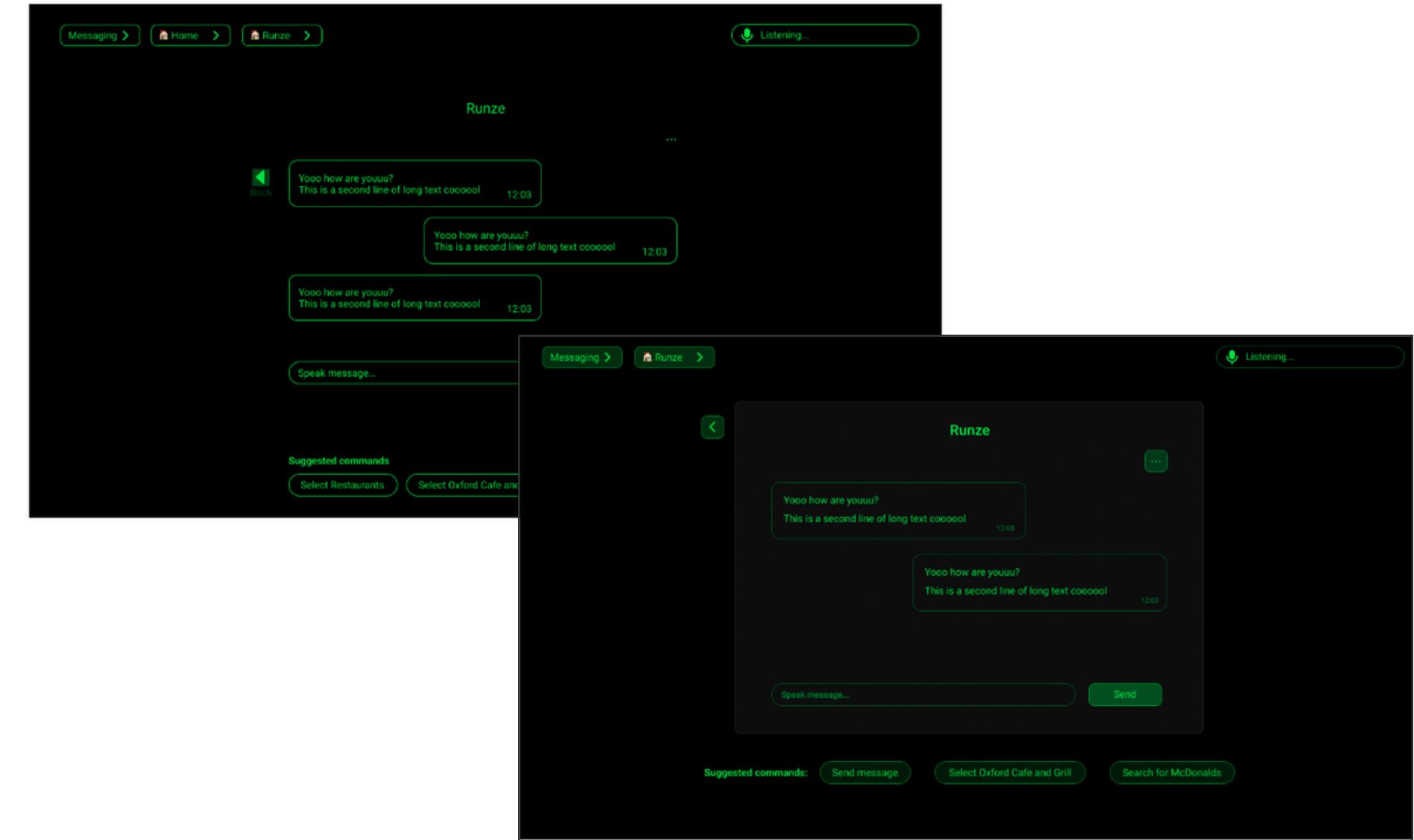
Layout allows **seamless multitasking** between physical and digital tasks

Consistent layout in all Glassify apps contributes to **user familiarity** and **reduces cognitive load**

Discussion

Interface design

- Prioritisation framework
- Layout transformation
- **Visual hierarchy**



Interaction design

- Voice-friendly shortcuts
- Special voice commands
- Error handling techniques

Hierarchy guides user attention effectively, enhancing **usability** and task performance

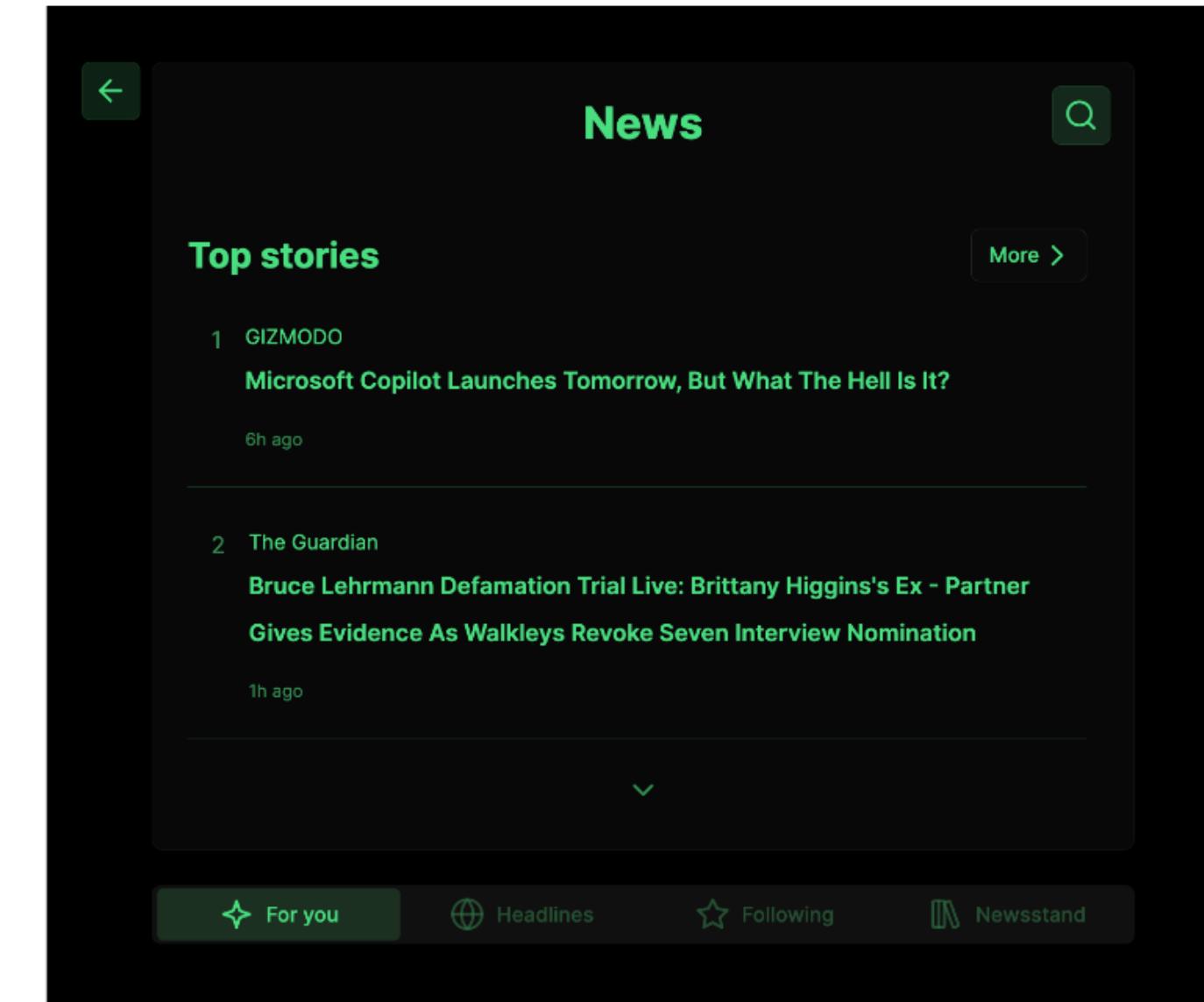
Discussion

Interface design

- Prioritisation framework
- Layout transformation
- Visual hierarchy

Interaction design

- **Voice-friendly shortcuts**
- Special voice commands
- Error handling techniques



Voice-friendly shortcuts “make up” for voice being **slower** than touch interactions on phones

Discussion

Interface design

- Prioritisation framework
- Layout transformation
- Visual hierarchy

Interaction design

- Voice-friendly shortcuts
- **Special voice commands**
- Error handling techniques

Microsoft Copilot Launches Worldwide Tomorrow, but What the Hell Is It?

It's kind of like if Clippy went to get his MBA and wants to be your personal assistant.

October 31, 2023 by Maxwell Zeff

Microsoft's generative AI work assistant Copilot will debut to millions of enterprise customers Wednesday. Chief Information Officers around the country are stepping up training in preparation for the Microsoft 365 product launch, according to the Wall Street Journal, and it may be a change coming to your office place as well.

Listen

Voice commands that consolidate multiple actions for a **well-integrated system**

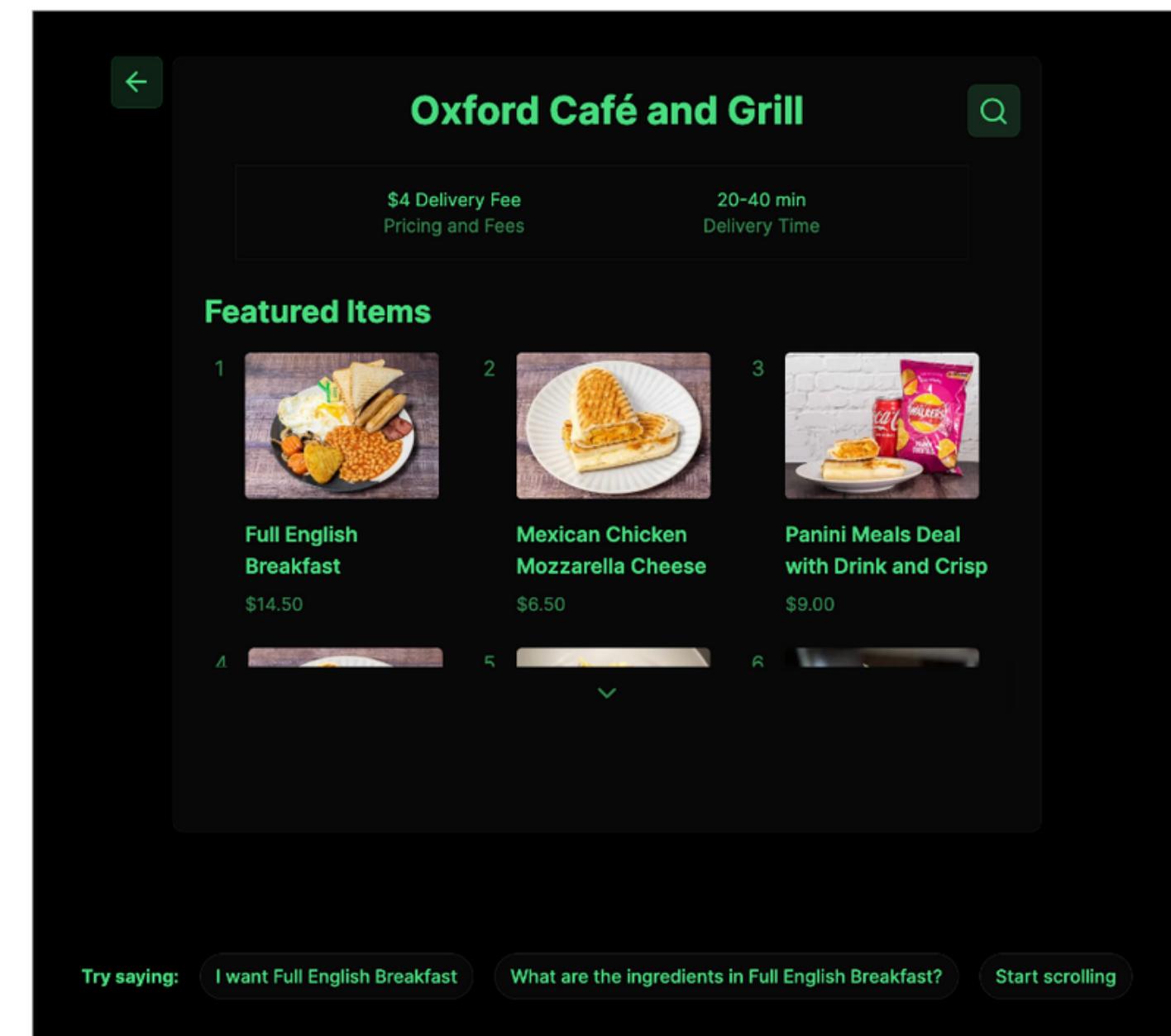
Discussion

Interface design

- Prioritisation framework
- Layout transformation
- Visual hierarchy

Interaction design

- Voice-friendly shortcuts
- Special voice commands
- **Error handling techniques**



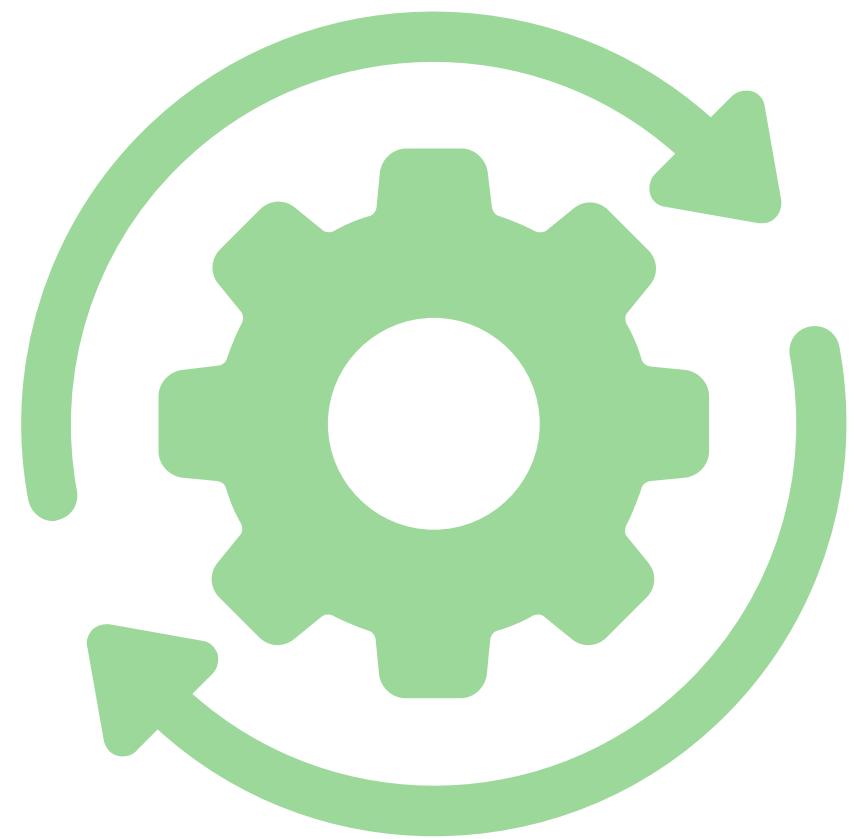
Tailored suggestions
Proactive clarification prompts
System status display

Agenda

- Motivation
- Research Problem
- Related Work
- Glassify Prelude
- Methodology
- Evaluation
- Discussion
- Future Work**
- Demo
- Conclusion

Future Work

1. Automating UI transformation
2. Incorporating gaze
3. Allowing one-shot commands



Demo

Glassify: A Retrospective Overview

Approach

- OHMD interface design
- Voice interaction design

Glassify: A Retrospective Overview

Approach

- OHMD interface design
- Voice interaction design

My contributions

- 5-step **transformation pipeline** backbone
- Plan and conduct **design workshops** and analyse findings
- Create UI prototypes and conduct **iterative evaluations** to enhance guidelines
- Implement **OHMD frontend UI** system and main controller
- Plan and conduct final **comparative study** and evaluate findings

Glassify: A Retrospective Overview

Approach

- OHMD interface design
- Voice interaction design

Results

- Reduces distraction
- Reduces mental workload
- Increases usability

My contributions

- 5-step **transformation pipeline** backbone
- Plan and conduct **design workshops** and analyse findings
- Create UI prototypes and conduct **iterative evaluations** to enhance guidelines
- Implement **OHMD frontend UI** system and main controller
- Plan and conduct final **comparative study** and evaluate findings

Appendices

OHMD Frontend UI

Glassify

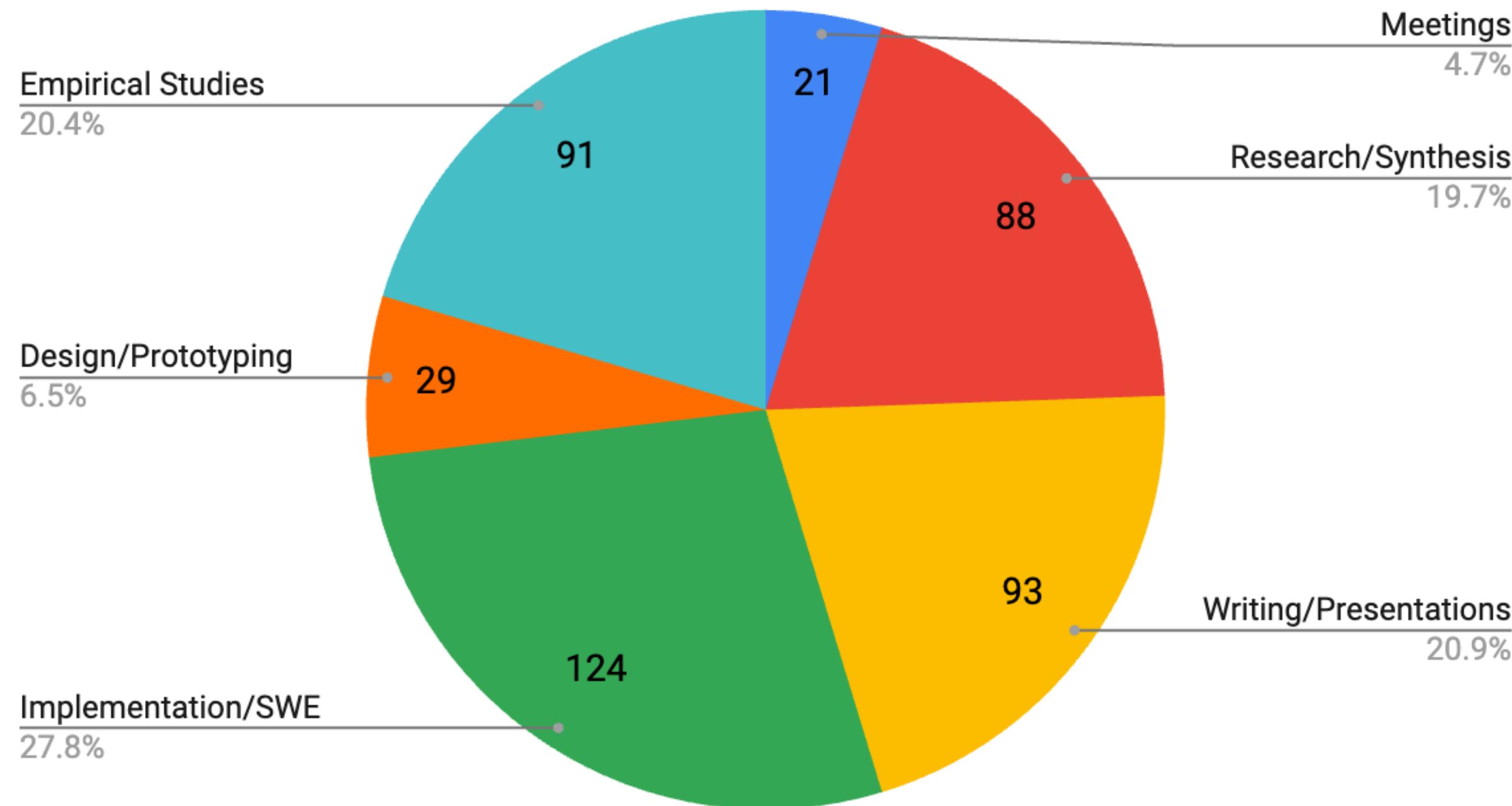
Glassify is a system to transform smartphone UIs into OHMD experiences.



vercel.app

Access the OHMD frontend deployment here: <https://ohmd-voicify.vercel.app/>

Hours spent on each task type



Task tracker + Gantt chart (Total 446 hours)

<https://docs.google.com/spreadsheets/d/1T7aOnfmMBw19y4atG90dBKTEgZnHVBY7dOHNO3UMGJs/edit?usp=sharing>

Prioritisation framework in action

Table 4.1. Matrix of UI element categories for task types. Examples are based on the Uber Eats homepage.

UI Elements	Primary Task	Secondary Task
	Selecting a restaurant to order from	Searching for restaurants
Critical	Show: Restaurants list, restaurant name	Minimise: Search bar
Helpful	Minimise: Delivery fee, delivery time, cuisine type	Hide: Filtering, sorting options
Non-essential	Hide: Restaurant preview images	Hide: Excessive filtering animations

Comparative study results

Table 7.1. Mean SUS survey scores per system from the comparative study.

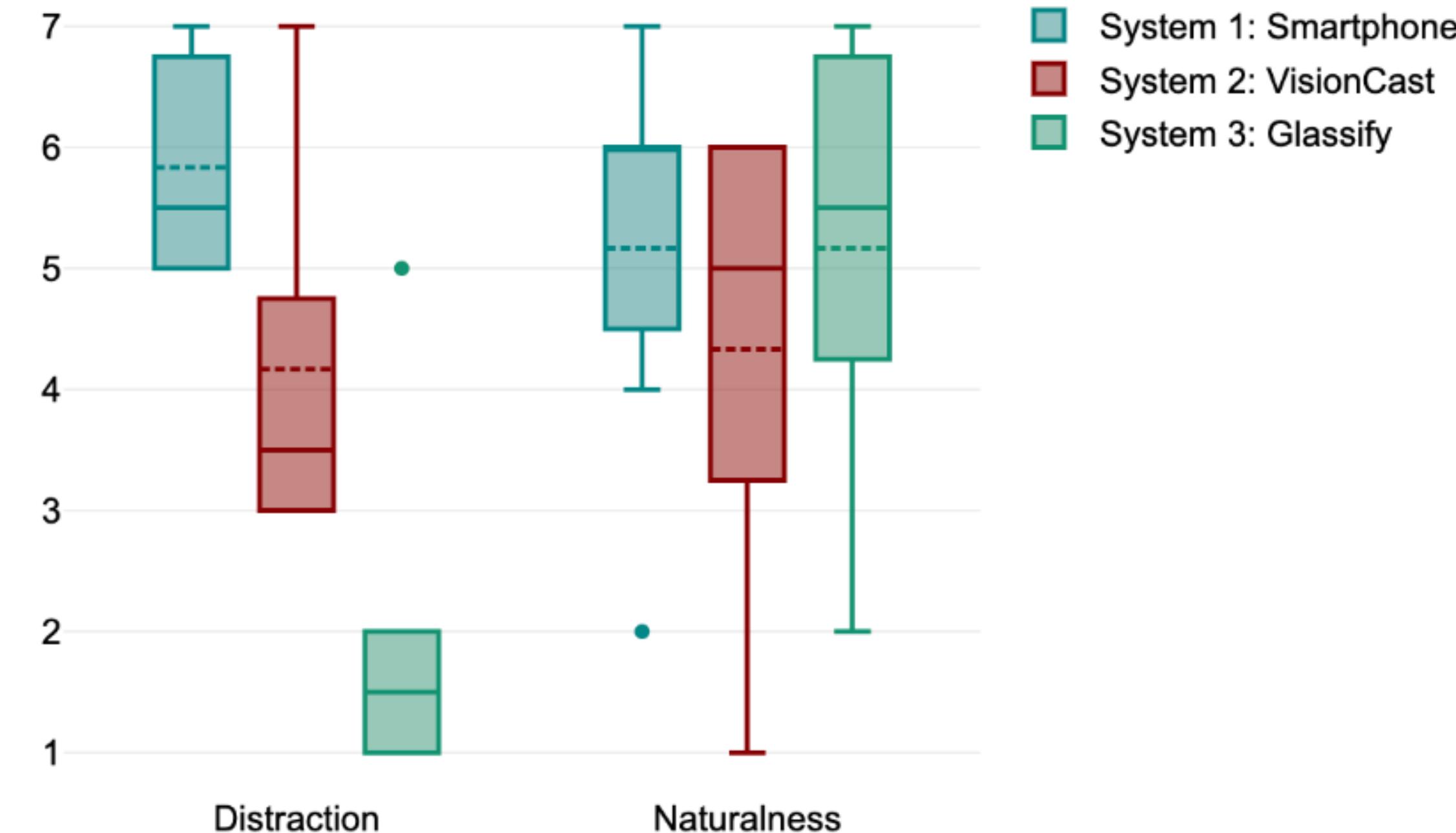
System	System Usability Scale (SUS) score
Smartphone	70.42
VisionCast	63.75
Glassify	74.17

Table 7.2. Mean Mental Workload scores per system from the comparative study.

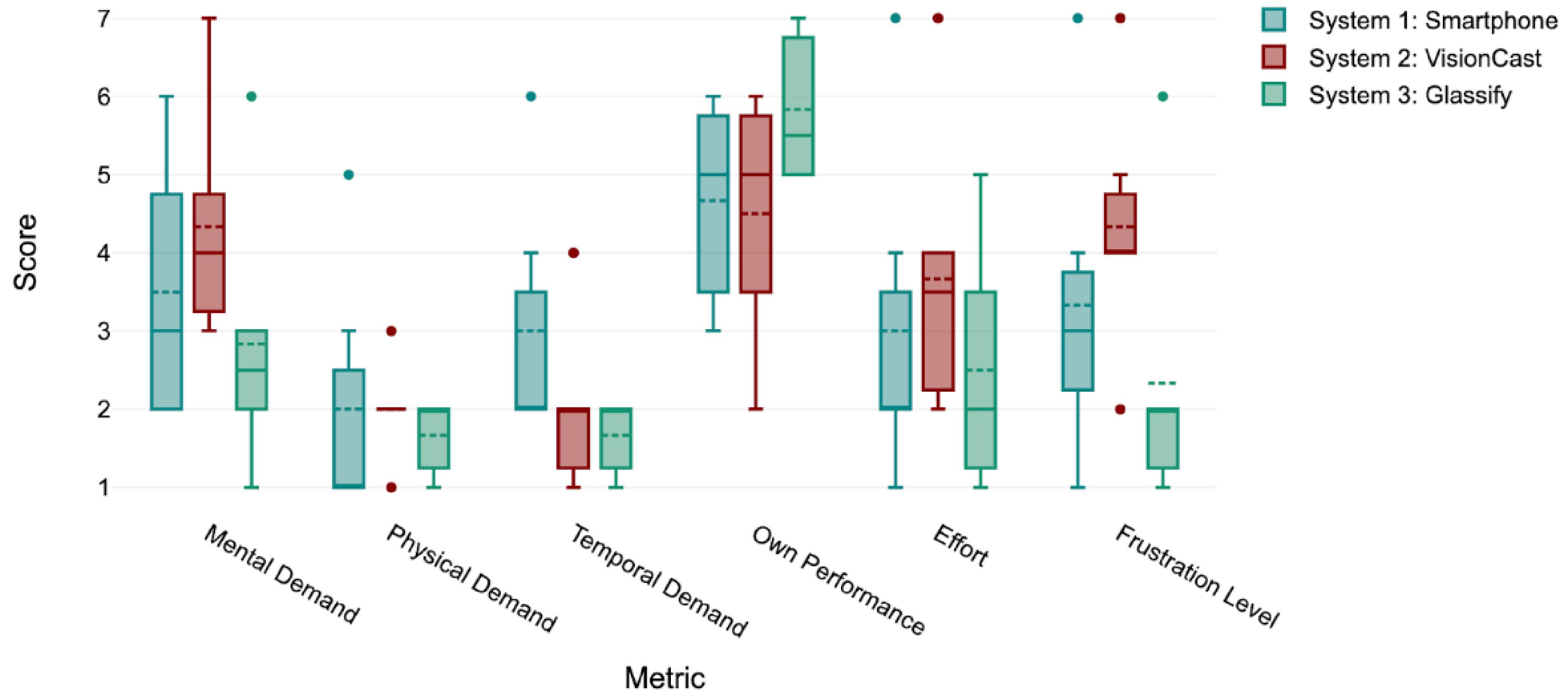
System	Mental Workload (RAW-TLX) score
Smartphone	42.92
VisionCast	47.08
Glassify	30.42

Table A-2.2. Mean scores for each metric in RTlx per system on a modified 7-point Likert scale from the comparative study.

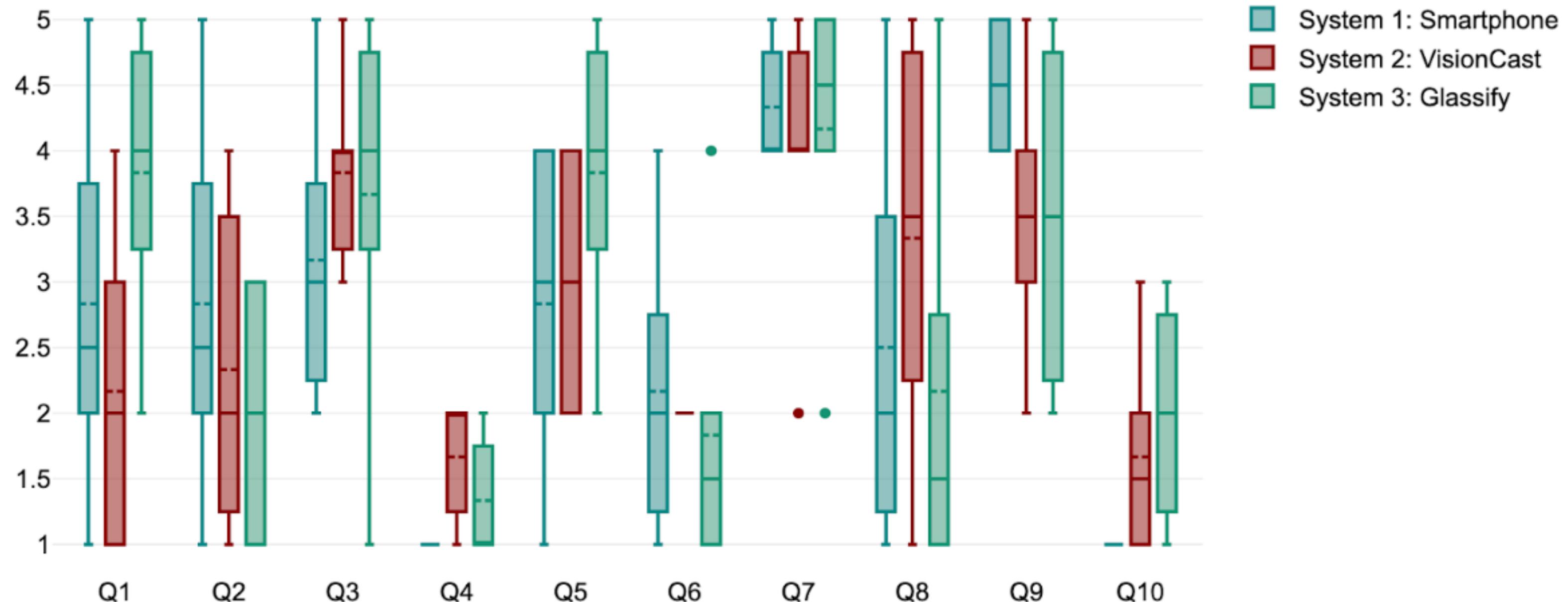
System	Mental Demand	Physical Demand	Temporal Demand	Own Performance	Effort	Frustration
Smartphone	3.5	2	3	4.67	3	3.33
VisionCast	4.33	2	2	4.5	3.67	4.33
Glassify	3.83	1.67	1.67	5.83	2.5	2.33



Raw-TLX



System Usability Scale (SUS)





Me with the XREAL Light

Thanks to Prof **Shengdong Zhao**
and Prof **Chunyang Chen**,
and my teammates **Runze Cai**,
Han Wang and **Dustin Vu**!