Improving the Modelling of Human-centric Aspects of Software Systems: A Case Study of Modelling End User Age in Wirefame Designs

Aria YukFan Jim[1], Hyun Shim[1], Jue Wang[1], Lionel Richie Wijaya[1], Rongbin Xu[1],
Hourieh Khalajzadeh[2], John Grundy[2] and Tanjila Kanij[2]
[1]*Faculty of Information Technology, Monash University, Melbourne, Australia*
[2]*HumaniSE Lab, Monash University, Melbourne, Australia*

Keywords: Human-centric Aspects, Modeling Frameworks, Wireframes, End-users' Age, Human-centred Design.

Abstract: Taking into account the diverse human aspects – gender, age, emotions, personality, language, culture, physical and mental challenges, etc – is critical towards achieving more human-centric design of software systems. Human-centric aspects affecting software have long been underestimated or even ignored as a result of the lack of in-depth capture and understanding during development. The use of technology has become the norm and the range of users has increased from just adults to children as well as seniors. Modelling frameworks are methods to represent the way a software system should be defined, and to date, little research has been done on age-related issues within modelling frameworks. In this paper, we investigate how human-centric aspects regarding age can be better modelled by extending these modelling frameworks. We introduce an extension to wireframe-based designs so that they can cater for decisions regarding age within the modelling framework. We have evaluated this modelling extension using multiple questionnaires as well as usability testing by using the extended age-modelling wireframe approach to design a news app. Questionnaires were used to evaluate the requirements of the users and developers for the extended wireframes. Our analysis shows that when using our extended wireframes, developers can cater for different user types and their accessibility needs easily and therefore users can use the prototypes with more ease.

1 INTRODUCTION

According to the internet usage report from Pew Research centre in 2019, 73% of United States adults over the age of 64 access the internet (Pew Research Center: Internet, Science & Tec, 2019). The average life expectancy of many high-income countries in the world such as Australia, Japan, United States is over 75 years (M. Roser and Ritchie, 2019). There is an increase in the average age of internet users which provides evidence that there is a need for better catering for elderly users of software systems. Furthermore, a study by Joyce and Nielsen in 2019 shows that teens, growing up with technologies, are also complaining about poor visual designs, such as font size, background colour and layout of certain websites (Hussain et al., 2017). The study also compares web-browsing behaviour between different age groups which indicates that users' age should be taken into account when designing software or websites.

When handling human-centric aspects in software design, age being just one example, developers need to carefully consider the limitations and abilities of these end users as well as their working environment (Kulyk et al., 2007). *"Elderly users can face issues such as screen readability due to visual impairments"* – which is not usually an issue for the mostly young developers who design and develop most software (Parker Software, 2019). Conventional modelling languages such as the Unified Modelling Language mostly just model technical requirements and (non-human-centric) non-functional requirements. *Human-centric aspects* of end users of software are one of the most significant factors to the success of a software system. However, these have been largely ignored and not modelled properly during the system development process to date (Grundy et al., 2020).

In this paper, we seek to improve the modelling process to better meet the human-centric issues regarding age. Our objective is to critically analyse the existing models, extend these models, and eventually

Jim, A., Shim, H., Wang, J., Wijaya, L., Xu, R., Khalajzadeh, H., Grundy, J. and Kanij, T.
Improving the Modelling of Human-centric Aspects of Software Systems: A Case Study of Modelling End User Age in Wirefame Designs.
DOI: 10.5220/0010403000680079
In *Proceedings of the 16th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2021)*, pages 68-79
ISBN: 978-989-758-508-1

build a prototype. For the evaluation, we will identify whether developers are easily able to create software systems with better usability for all age groups as well as whether the different user types are able to easily use the software systems. Our research aim is to investigate how human-centric aspects regarding age can be expressed in current modelling frameworks, and to this aim, we focus on three key research questions:

RQ1. Which Modelling Frameworks are Best Fitted for Adding Human-centric Aspects Regarding Age?– For example, use case diagrams, class diagrams, wireframes, etc. To determine whether creating a new modelling framework or extending an existing one will be better, and to find out which one is the most extendable if we choose the later. To do this, we have gathered results from a developer survey and relevant research papers.

RQ2. What Progress Can Be Made on Human-centric Aspects Regarding Age for Requirements Collection and Modelling? – How can we best model user characteristics regarding age in software requirements and/or design models? For example, by enlarging fonts based on age, simpler user interfaces for elderly people, explanation on certain keywords, etc. We need to define the age range for each of the age groups as well as identify human-centric aspects that can be improved for each of the age groups.

RQ3. What Application Domains Would Benefit from Human-centric Aspects Regarding Age? – For example, common applications such as news apps, discussion forums, social media, etc. We wanted to find out whether there are certain domains that can best use or extend our new modelling framework, and to find any patterns or reasons that make this so.

Section 2 briefly explains the methodologies used throughout the research, and provides an overview of the background for choosing age as the main objective among all the human-centric aspects for this study. We present our approach in Section 3 and our evaluation approach in Section 4. Section 5 presents our extended modelling framework including examples of the artefacts resulting from the research. We also discuss evaluation responses from our developer survey questionnaires. Section 6 provides a summary of the research papers that are related to our study. We will finally conclude the paper in Section 7.

2 METHODOLOGY

We reviewed a number of research papers related to extending modelling languages to support the modelling of different human-centric aspects, including modelling emotions, age, culture, language etc (Curumsing et al., 2019; Kamalrudin et al., 2012; Spichkova et al., 2015; Alves et al., 2020). We investigated how the languages were extended and which human-centric aspects were modified in these existing research papers. We also reviewed various works that extended existing modelling languages to capture additional non-functional characteristics (Goncalves et al., 2018; El-Attar et al., 2015; Grundy and Patel, 2001). We investigated what aspects could be changed in the existing modelling languages, in order to capture information about different age groups of end users. After our analysis, we developed a set of extensions to the widely used wireframe based design notation (de Lange et al., 2020) in order to model different end user ages, age-related implications on the design, and different design decisions based on user age characteristics.

To evaluate our extended design modelling notation we used a set of different questionnaires, aimed at different classes, i.e., users and developers. We provided multiple solutions to modelling software design decisions regarding human-centric aspects related to age. We wanted to ask target end-users to identify the ones that are good for the age group, in their opinion. We used a range of target end users from different age groups to evaluate a prototype based on our augmented wireframe design models. Using their feedback, we checked to see which solutions are the best fit for our prototype use cases.

Our extended wireframe modelling frameworks are to be used by Software Engineers to create an end-product that is more age-aware and supports different interfaces and interface components for differently-aged end users. We prepared a different set of questions to ask a range of developers their opinion on the new modelling framework. We wanted to identify key current issues that can be found with the pre-existing model frameworks not supporting modelling of end user age. We also wanted their opinion on the new framework and whether it helped them in addressing these issues. Using the questionnaire results, we have refined our extended wireframe-based design approach to better match the needs of the developers.

3 OUR APPROACH

There were multiple modelling frameworks that could be extended to support end user age difference modelling in our research. We decided that extending wireframe-based design models would be the best option as most of the changes required for the different age groups were based on the application user interface. Wireframes provide a simple way for developers to create the basic design of an application and our extension point was to create a workflow that a developer can follow for each of the different age groups. In recent years, various other works have looked to extend the use of wireframe-based UI design and to suggest wireframe-based designs or partial designs (de Lange et al., 2020; Chen et al., 2020). None have focused on supporting different design decisions based on end user ages.

Based on the research review, our age groups are defined as: *1) Kids;* Under 15 *2) Adults;* Aged between 15 and 64; and *3) Seniors:* Aged above 64. To model different ages, age-related design considerations, and age-related design differences to wireframe models, we added three features to the original wireframe modelling framework:

1. Branching (Branch a screen into two or three pathways)
2. Merging (Merging two or three pathways into a single pathway)
3. Splitting (Splitting a screen into multiple screens)

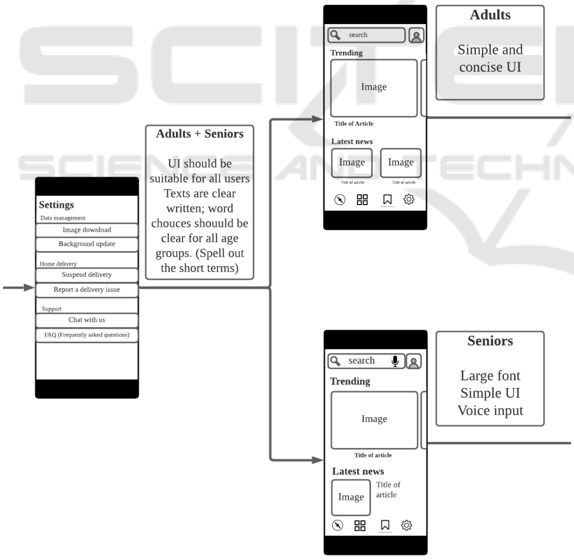


Figure 1: Example of branching.

Figure 1 shows an example of branching. A single pathway has been branched into two pathways. This is used when there is a need for different user interface or functionality for the same screen for different aged users. It thus gives developers a tool to express the need for a change in a single screen for different ages.

Figure 2 shows an example of merging. Two pathways have been merged into a single pathway. This is useful when multiple pathways arrive at a specific screen that will be the same for multiple age groups that have been previously branched out. It saves the developers from creating the same screen design for different pathways. It also helps show that multiple screens will be the same across different age groups.

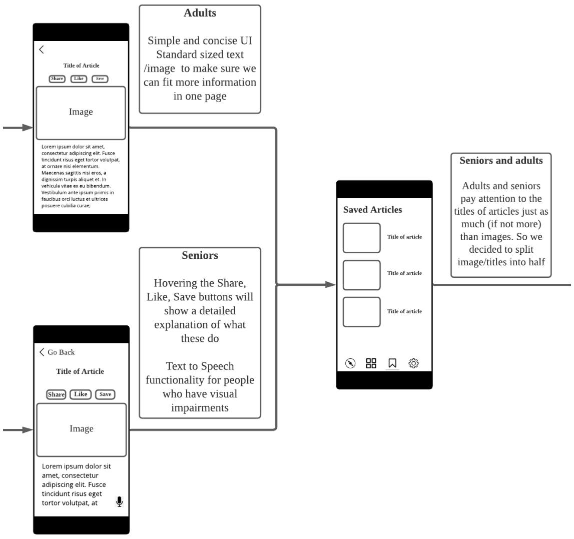

Title of Article
Share
Like
Save
Image
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce tincidunt risus eget tortor volutpat, at ornare nisi elementum. Maecenas sagittis nisi eros, a dignissim turpis aliquet et. In vehicula vitae ex eu bibendum. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae;
Adults
Simple and concise UI Standard sized text /image to make sure we can fit more information in one page
Seniors
Hovering the Share, Like, Save buttons will show a detailed explanation of what these do
Text to Speech functionality for people who have visual impairments
Go Back
Title of Article
Share
Like
Save
Image
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce tincidunt risus eget tortor volutpat, at
Saved Articles
Title of article
Title of article
Title of article
Seniors and adults
Adults and seniors pay attention to the titles of articles just as much (if not more) than images. So we decided to split image/titles into half

Figure 2: Example of merging.

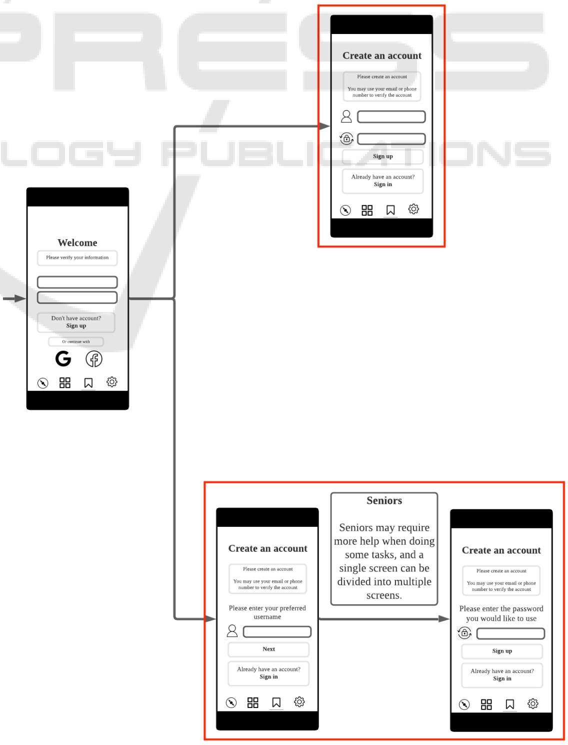

Welcome
Please verify your information
Don't have account?
Sign up
Or continue with
Create an account
Please create an account
You may use your email or phone number to verify the account
Sign up
Already have an account?
Sign in
Create an account
Please create an account
You may use your email or phone number to verify the account
Please enter your preferred username
Next
Already have an account?
Sign in
Seniors
Seniors may require more help when doing some tasks, and a single screen can be divided into multiple screens.
Create an account
Please create an account
You may use your email or phone number to verify the account
Please enter the password you would like to use
Sign up
Already have an account?
Sign in

Figure 3: Example of Splitting.

Figure 3 shows an example of splitting. A single screen is split into two screens. This is useful for age groups that require additional assistance in getting a task done. For example, adults may prefer a single register page as it allows them to quickly register. However, kids or seniors may need additional help when registering, e.g. by using a step by step multiple screen approach. Using multiple screens will give developers more room to explain key features within each screen, make items larger, or choose different interaction components. This will be a trade-off between speed and ease-of-use.

4 EVALUATION APPROACH

4.1 Prototype News App

We created a prototype news app based on our extended wireframe approach to evaluate how easy it was for a developer to use this extended wireframe model. The prototype was also used to evaluate the final prototype system by different age group target end-users. We chose a news app as the prototype as (i) most age groups use them or are interested to use them; (ii) despite their seaming simplicity, they often have quite complex interfaces and design decisions are not always fitting differently aged users; (iii) the interface design is reasonably detailed but not overwhelmingly so; and (iv) many lessons from news app development can be applied to social media, communications, education and other widely used apps (Constantinides et al., 2015). Hence, the news reading app domain contains enough variation points between the three age groups to try out the use of our extended wireframe modelling language features.

We used our extended wireframes to design and develop a Figma-based prototype. The extended wireframe designed to create the news app prototype is shown in Appendix. By using the extended modelling framework we captured key human-centric aspects regarding the age of the users and designed solutions to accommodate these. This news application has different functionality and user interface for different age groups of its users. In this prototype, we decided to keep the colours for each of the branches specific to the age groups even if a screen was for all three user types. For example, a screen that has not been branched out which is used for all three user types will still have different colour themes to allow for a more consistent experience for the end user.

In the extended wireframe design model, we have many common pages for all three age groups. We needed to find a balance point for these age groups such as use of font size, user interface complexity, and any related points to ensure that the design for the common pages is clear enough for use from all three age groups. An age selector is presented to users at the start, as shown in Figure 4.

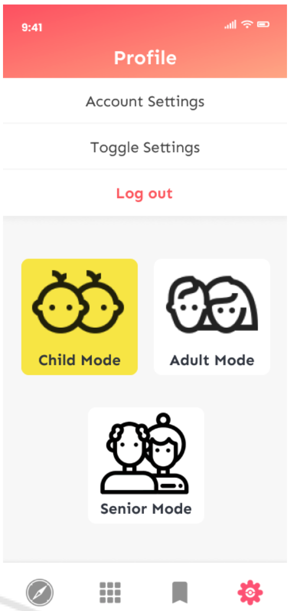


Figure 4: Age category selection page.

4.1.1 Children

For the **kids application**, the key aim was to make the application enjoyable and attractive enough for the children. Some user interface design decisions that needed to be considered are:

- *Colour:* Research showed that children prefer colours like red and yellow and thus we used a rainbow gradient colour schema to catch the eyes of the children.
- *Font Style:* Use a fun looking font size to help children stay focused.
- *Icon Selection:* We chose a more cartoonish icon for children to cater for their shorter attention span.

Figure 5 shows a user interface designed for kids, based on that specified in our extended wireframe.

4.1.2 Adults

For the **adult application**, the key aim was to make the app clear and simple so that they will be able to quickly navigate between pages without too much overhead information. Thus key design decisions for the adult app user interface design included:

- *Font Size:* Normal size.

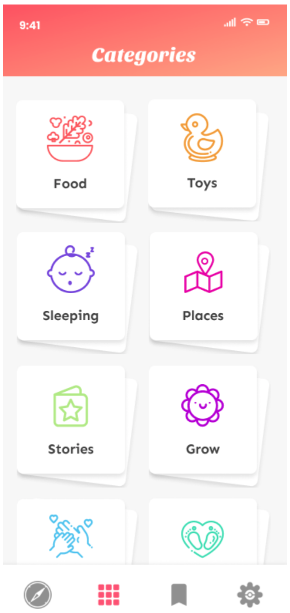


Figure 5: User interface for kids adapted from the extended wireframes.

- *Colour:* Simple and conventional colours. More professional compared to the kids version.

Figure 6 shows a user interface designed for adults, using our extended wireframe.

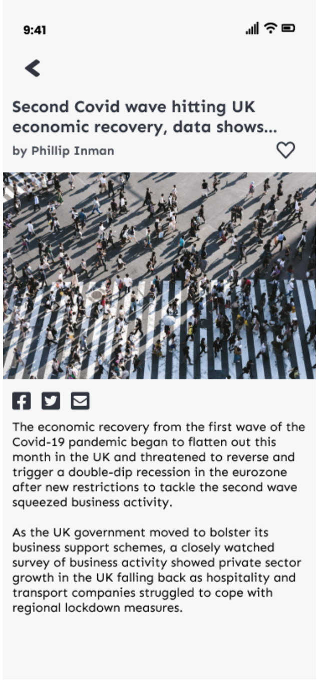
4.1.3 Seniors

The **senior application** is similar to the adult application, but we needed to make sure that everything is large enough, easily accessible, and that the user interface is simple enough. We added some additional functionality to help aid seniors so that they can comfortably use aspects of the technology that they may not be familiar with. Some key design decisions for the senior age group include:

- *Colour:* Similar to adults, use simple and conventional colours.
- *Font Size:* use a larger font size to aid visual impairments.
- *Voice Control:* Senior users may have trouble typing on a mobile device or reading smaller font sizes and speech-to-text and text-to-speech functionality would be good for them.
- *Larger Components:* Make items such as buttons, and links large enough for them to easily click.

Figure 7 shows a user interface designed for seniors, using our extended wireframe.

We also provided a toggle setting, shown in figure 9, so that users can customise the app styles such as the font size, font style, etc. This was necessary because there may be some users who do not require the customizations that were categorised within their user age groups. Other customizations that were de-



Second Covid wave hitting UK economic recovery, data shows...

by Phillip Inman

The economic recovery from the first wave of the Covid-19 pandemic began to flatten out this month in the UK and threatened to reverse and trigger a double-dip recession in the eurozone after new restrictions to tackle the second wave squeezed business activity.

As the UK government moved to bolster its business support schemes, a closely watched survey of business activity showed private sector growth in the UK falling back as hospitality and transport companies struggled to cope with regional lockdown measures.

Figure 6: User interface for adults adapted from the extended wireframes.

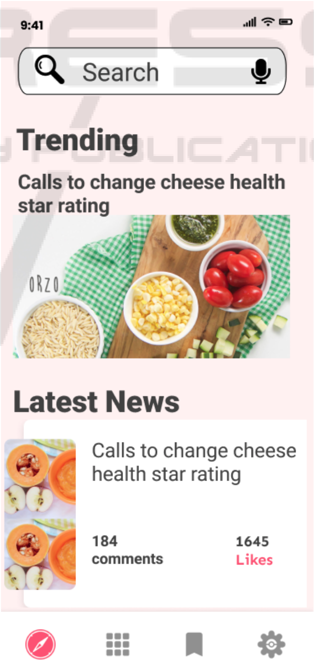
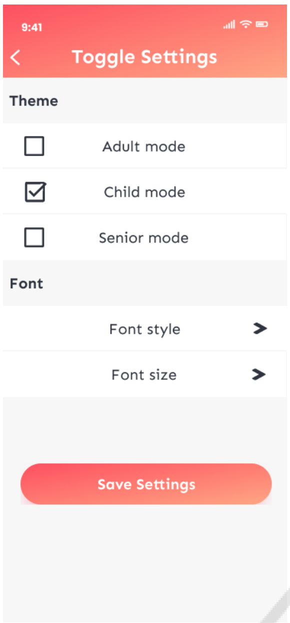

9:41
Search
Trending
Calls to change cheese health star rating
ORZO
Latest News
Calls to change cheese health star rating
184 comments
1645 Likes

Figure 7: User interface for seniors adapted from the extended wireframes.

termined necessary during our research are specified in the extended wireframes. This allows developers to choose whether to include a customization within their applications.



9:41

Toggle Settings

Theme

□ Adult mode

☑ Child mode

□ Senior mode

Font

Font style >

Font size >

Save Settings

Figure 8: Toggle setting.

4.2 Usability Evaluation

To evaluate the news app prototype's usability, a usability test was conducted to gain insights into what went well and what needs to be improved in our extended wireframe modelling approach. Users were asked to complete two tasks:

- Change the user mode along with some settings to match their preferences; and
- Find an article they are interested in, save the article, and open the article again using the saved articles page.

The details of the tasks given to the users were purposely given without too much detail on how to navigate within the application. This was so that we can see how a new user will act when using this application. These tasks helped us to gather user data on key age-based design decisions in the news app:

- Whether each user type is able to use the interaction flow designed for them within the application, as well as change the settings to match their accessibility needs; and
- Whether the navigation supported within the application is easily understandable to new users.

The testers were asked to record whether the tasks were successfully completed as well as to describe any troubles that they encountered during performing the tasks. We conducted a preliminary test, and using the results, we enhanced our extended wireframe models to increase the usability of the prototype. We plan to conduct a more comprehensive usability evaluation with developers to evaluate whether the extended wireframe is applicable to different domains and can be easily used to create prototypes. We also plan to conduct usability evaluations with end-users from various age-groups to evaluate whether the created prototype actually increased the usability of the news app prototype for each age group.

5 RESULTS

In this section, we present the results of our survey of developers and users. We collected a total of 27 responses – 6 from developers, 21 from app end users.

5.1 Evaluation Results: Developer Questionnaires

Six developers responded to our survey – 3 with 5 to 10 years development experience, 2 with 1 to 4 years, and 1 with more than 20 years of experience. Among all of these developers, 4 said they think there is a need for human-centric aspects regarding age in modelling frameworks e.g UML diagrams, Wireframes, User stories, etc. One had never though of this issue, and one though it is not important. We collected a list of approaches and frameworks they currently use to deal with age related issues of the users in the software that they develop. This included use of wireframes (3 of the developers), UML (3), user stories (2), use cases (2), and BPMN diagrams (1).

We asked the 6 developers whether they have encountered any issues regarding age during software development. Based on their responses, as shown in figure 9, one of the key issues is that the majority of developers do not address differing age of their end users at all in their software development. To address this issue in this study, we captured a lot of end user requirements from our end user survey questionnaires. The second issue is that some developers believe it is hard to define an "age group". To address this, we decided to research more about the target audience of the particular software and then defined the age groups accordingly.

We asked developers what tool(s) they used to help them address differing age issues of their app and web site end users. They mostly stated they never used any tools. Some suggested that although they

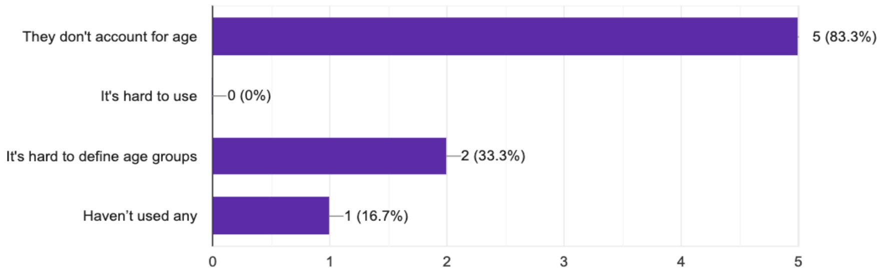


Figure 9: Developer responses on designing in issues regarding different end user age during development.

did not use any tools in particular, they tried to address age related problems such as security restrictions in the identity server (e.g. some websites are for adults only); and informally captured non functional requirements in text, annotate the odd use case/user story. For a news application, age selection is important to restrict inappropriate news from the children. Icons, text, and background colours might also need to be changed to accommodate different types of users sight limitations and preferences based on their age.

Regarding what other human-centric aspects they think could be a good addition to better support in software modelling frameworks, developers included gender, culture, end user language, physical and mental challenges of users, accessibility needs of end users, Convenience of usage, usability test (user test), cognitive load (mental effort) test (including performance measures in terms of user's response time taken to complete a task and its accuracy, or physiological measures, e.g. pupil dilation and blink rate).

5.2 Evaluation Results: End-user Survey

We received 21 responses, 13 from 18-25 (Young Adults) respondents, 6 from 26-49 (Adults), and 2 from those 50-64 years old. The age demographics for the questionnaire participants are mostly young adults with ages ranging from 18 to 25. We did not have any younger participants (below 18) and senior participants (above 65). Most of the participants used technology for daily activity for a significant amount of time (8 hours+). A majority of the participants found that using technology is easy for them (14 people said they are proficient with it, five said they are good with it, and one said they are doing okay with it). We also found out that the most popular device is a smartphone used for social media, entertainment and news apps. We asked if any participant had any accessibility requirements. The majority of people saying they had visual impairments, such as needing glasses to read, having short sight, and one person answered that they needed to use voice output to "read" app text.

5.3 Research Question Answers

RQ1. Which Modelling Frameworks are Best Fitted for Adding Human-centric Aspects Regarding Age? – From the result in figure 10 we can see that most of the problems are User Interface related issues. This is one of the main reasons we decided to extend wireframes since wireframes provide early visuals that can help with these problems, they are also easy to adapt compared to conceptual designs.

RQ2. What Progress Can Be Made on Human-centric Aspects Regarding Age for Requirement Collection? – We used questionnaires for collecting data. We created both end user and developer questionnaires to capture data from multiple angles. We got a range of feedback from different age groups that indicated usability enhancement in the prototype news app for different age groups of target end users. However, we need to recruit more participants to gain a larger sample size, especially younger (under 18) and older (over 65) end users.

RQ3. What Application Domains Would Benefit from Human-centric Aspects Regarding Age? – According to our survey results in figure 11, we can see that our participants are commonly using a range of different application domains such as social, entertainment, study, etc. These results helped us in deciding on which domain our example extended wireframe and prototype would focus on. Generalising our experiences to those other common domains would help to show if the approach can enhance different age group usability for them as well.

5.4 Discussion

We chose to extend wireframe design models with information about the age of target end users and alternative user interface design decisions based on these ages. However, other modelling frameworks such as interaction and sequence diagrams may help developers understand the human-centric aspects regarding age further and use similar approaches based on the wireframe extensions we have created. For ex-

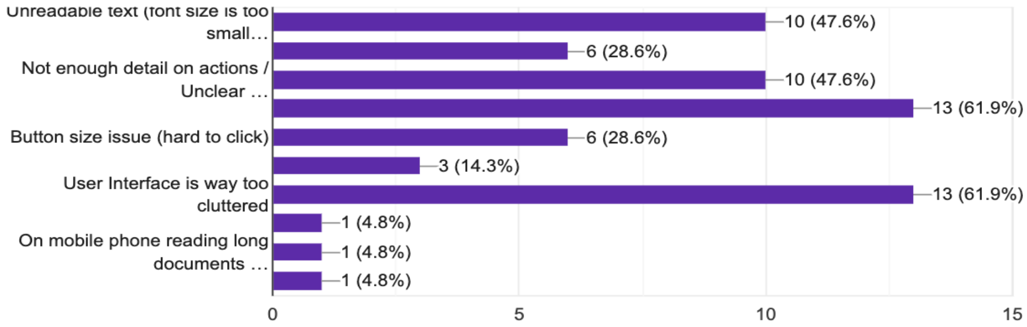


Figure 10: Key pain points of using software systems.

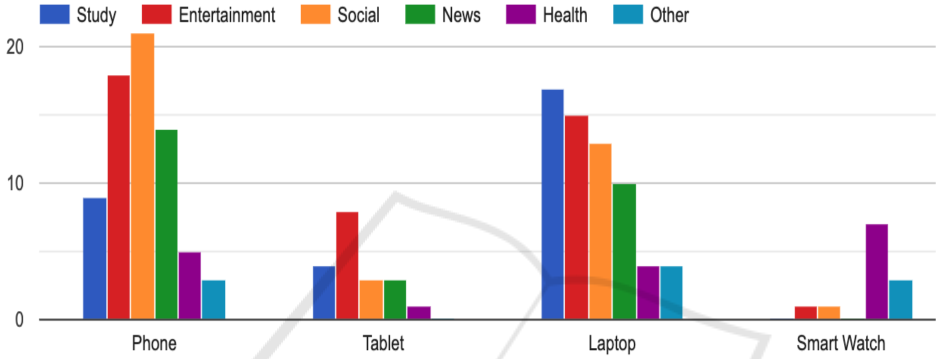


Figure 11: Devices used by participants and for what purposes.

ample, figure 12 contains an interaction diagram that is adapted from the extended wireframes presented in this paper. Adapting other human-centric aspects, such as gender, emotion, and physical support, into the extended modelling framework presented in this paper may also be beneficial. Additional research can be done to increase the range of our user types and changes to the designs to accommodate their needs. Using approaches developed in the accessibility research field could also help us to better understand the needs of users with a range of disabilities. The specific requirements of these user types need to be identified so that the developers can create an appropriate pathway and design decisions for those users in our extended wireframe models.

6 RELATED WORK

We reviewed the key works done in three different categories related to designing for children, seniors and extending models to capturing emotions of end users. The age classification we used in this paper is called Life Cycle Groupings (Statistic Canada, 2017). According to Statistics Canada, age groups can be defined as: Children aged from 0 to 14 years; youth aged from 15 to 24; adult aged from 25 to 64 years; and senior aged from 65 and above. The Australian Bureau Statistics (ABS) uses a similar convention except they group both youth and adult as working-age population (Australian Bureau of Statistics, 2019), which we named it as adult throughout the research.

6.1 Children

Mobile Educational Applications are used by (Masood and Thigambaram, 2015) for usability testing. By recording with eye tracking glasses, they found some children had problems working with mobile applications. The system status is not apparent for them and they have a hard time finding out what to do next. They also had problems where they could not remember which page or button was accessed earlier. This children-oriented software needs to more clearly show the current state of the page, and sometimes the child users may need some guidance to do the next step. Help toolboxes and documentation were identified as necessary. Another important factor is to consider whether the buttons and menus are simple enough for children. Buttons and menu links should be easily identified as being clickable while items such as menu headings and titles should be easily identified as being not clickable.

A Fingerprint app (Pan, 2010) is used to describe

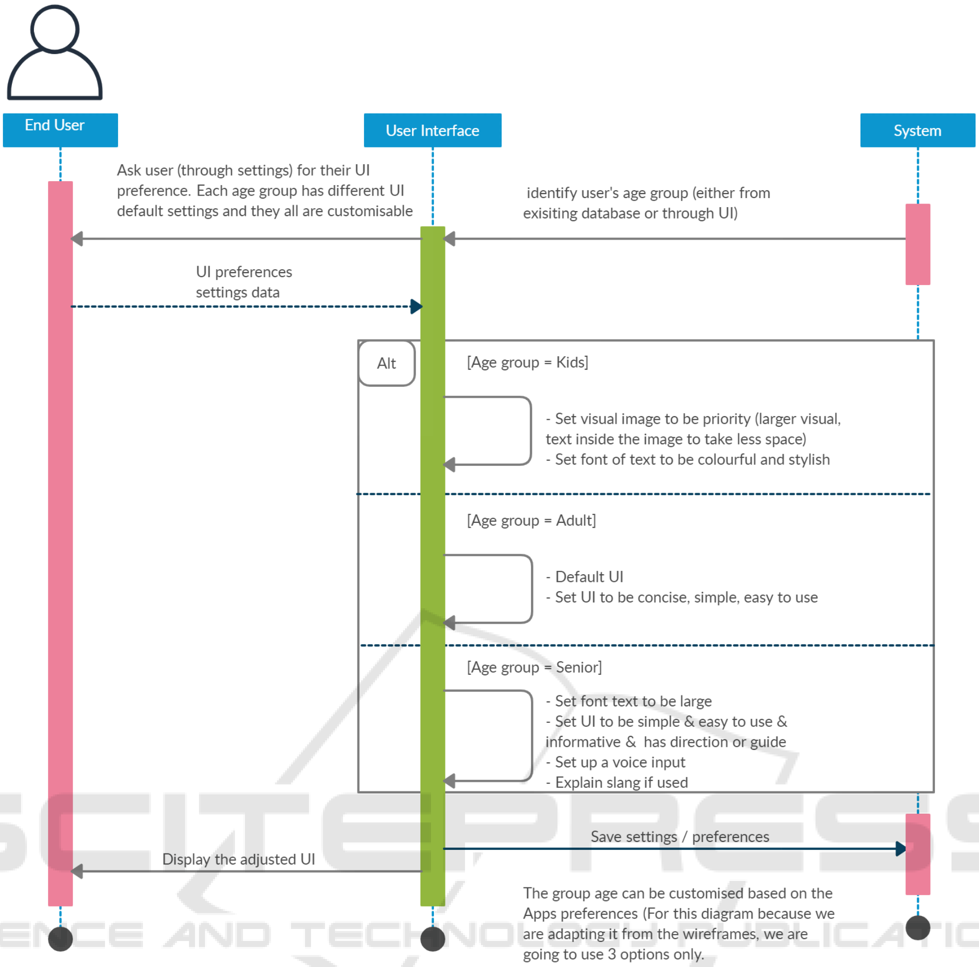

End User
User Interface
System
Ask user (through settings) for their UI preference. Each age group has different UI default settings and they all are customisable
identify user's age group (either from exisiting database or through UI)
UI preferences settings data
Alt
[Age group = Kids]
- Set visual image to be priority (larger visual, text inside the image to take less space)
- Set font of text to be colourful and stylish
[Age group = Adult]
- Default UI
- Set UI to be concise, simple, easy to use
[Age group = Senior]
- Set font text to be large
- Set UI to be simple & easy to use & informative & has direction or guide
- Set up a voice input
- Explain slang if used
Save settings / preferences
Display the adjusted UI
The group age can be customised based on the Apps preferences (For this diagram because we are adapting it from the wireframes, we are going to use 3 options only.

Figure 12: Interaction diagram adapted from the extended wireframes.

how to design the software user interface for the children. This work discusses four key points for the vision element regarding kids – integer vision effect, functional area design, icon and button design, and font design. For example, children may not understand the text inside a button but icons can be designed as buttons to show they are clickable.

The work of (Michaels, 1924) and (Boyatzis and Varghese, 1994) discusses colour preferences of children users as well as the effect of colour on children's emotions. They did colour tests with children around 6 years old, and found that their top three favorite colours are yellow, red, and green. Another type of testing they did was called the story test. If a child heard a happy story, most of them would select yellow. Using this research, they used yellow and red as the gradient colour for the theme colour, and made the whole app look funny and attractive.

6.2 Seniors

(Boll and Brune, 2015) provide a set of user interface design guidelines for people between 55 to 75 years old. They found 41 participants between 55 and 75 years to fill out questionnaires to research the actual requirement and the problems they have. They used the results to make the user interface design guidelines. According to the guidelines, for font design, the sans serif fonts provide good readability. For colour selection in the main content page, they recommended a light grey background with black fonts. Regarding the size of the icons and the buttons, most of the users reported that the icons are too small, and that double-clicking a small button is a problem for elderly people. The last important thing is the structure of the page. For example, in menu page, a good menu structure helps users to navigate through the user interface more easily. The menu needs to be put in conventional positions to make sure the position of the menus are consistent in the entire software.

(Curumsing et al., 2019) focuses on designing emotion-oriented software, based on the smart home device for elderly. They used extended Goal models, Interaction models, Scenario models, Role models, and Behaviour models to keep track of the "cared for" to a list of emotions. Analysis of these emotions helps developers to understand the expectations of an older adult using the smart home. Using this approach, a goal model for the smart home device was created. The model includes different emotions for the elderly people to help get the elderly people to accept the device and feel like this is what they need. The software can catch the emotion of the user, and analyse their expectations. Thus, the software can make corresponding responses to meet the needs of users. The software can understand users' emotions, this could become one of the determinants of software success.

(Curumsing et al., 2015) demonstrates a case study of an emergency alarm system for elderly people, presenting the entire suite of models for this case study. They suggested a few important factors on designing the framework and also keeping the interest of the elderly people. Firstly, in order to encourage elderly people to adopt technological solutions, they have to be designed in such a way that they suit the needs of its users, are easy to use, and cost effective. The second important factor determining the successful adoption of technology consists of willingness and enthusiasm for acquiring new knowledge. However, this is rarely the feelings expressed by elderly people when it comes to using a system which is linked with a stigma. For example, some refused to use the pendant because of its visibility to others. It was viewed as a sign of stigma and old age. These are the way users perceived technology from an emotional aspect.

(Wagner et al., 2014) explains the impact of age on usability. It states that there is an increase in the average age of internet users which provides evidence that there is a need for catering elderly users for applications. Usability helps organisations by improving the job performance, gaining higher productivity and reduced costs by the users. There are five conceptualisations of age; Chronological or calendar age, Functional or performance-based age, Subjective or psycho social age, Organisational age, and Life span concept of age. Currently, mainly research chronological age is used to allow for consistency and comparison with the existing literature. We will also be using chronological age as the other conceptualisations are mostly related to chronological age.

Finally, (Holzinger et al., 2008) aims to derive metric-based benchmarks for measuring usability. This study suggested two aspects of usability: passive and active interaction. Passive interaction means users are not directly interacting with it, but it helps out in the background. Active interaction means users directly interact with the technology. Several questions were developed to ensure it meets both passive and active interaction. They present an analogy between user anxiety and metrics. For our prototype, it will be a good metric to evaluate our product's trustworthiness and acceptance.

6.3 Emotions

An emotion orientated software development method was developed by (Chen et al., 2015). Existing emotion-aware applications lack accuracy in terms of emotion recognition due to the small scale of data collected. Through cloud-based computing and cloud assisted resources, mobiles phones can collect much more data. Combined with the architecture itself, applications can recognise user's emotional changes by big data analysis. Based on the user's current emotion, a common list of feedback is generated in the remote cloud. The information is transferred back to the local cloud, providing users with personalised services. Various in-home devices were used to accommodate the user's emotion. We could modify this approach and base the categorisation on age group. An application would analyse the habits of that specific age group in order to adapt the application to better supporting the user, based on their age.

7 SUMMARY

There is a need to incorporate human-centric aspects into modelling frameworks in order to improve the suitability of software produced for diverse end users. There are many existing modelling frameworks, and we expect that modifying existing modelling frameworks will be better than creating a completely new one. Most do not currently support modelling the age of end users and providing different design solutions for different age groups and needs of end users. We developed a set of extensions to the commonly used wireframe modelling approach to incorporate different designs for child, adult and senior end users. We evaluated our modelling approach with developers and a prototype news app developed using our approach with a range of differently aged end users. Further work includes incorporating other human-centric aspects into the extended wireframe model e.g. gender, culture, language, and trying the same model extension approach in other modelling frameworks, such as user stories, use cases, and sequence diagrams.

ACKNOWLEDGEMENTS

Support for this work from ARC Laureate Program FL190100035 is gratefully acknowledged.

REFERENCES

Alves, T., Natálio, J., Henriques-Calado, J., and Gama, S. (2020). Incorporating personality in user interface design: A review. *Personality and Individual Differences*, 155:109709.

Australian Bureau of Statistics (2019). Australian demographic statistics, jun 2019. *https://www.abs.gov.au/*.

Boll, F. and Brune, P. (2015). User interfaces with a touch of grey?–towards a specific ui design for people in the transition age. *Procedia Computer Science*, 63:511–516.

Boyatzis, C. J. and Varghese, R. (1994). Children's emotional associations with colors. *The Journal of genetic psychology*, 155(1):77–85.

Chen, J., Chen, C., Xing, Z., Xia, X., Zhu, L., Grundy, J., and Wang, J. (2020). Wireframe-based ui design search through image autoencoder. pages 1–31.

Chen, M., Zhang, Y., Li, Y., Mao, S., and Leung, V. C. (2015). Emc: Emotion-aware mobile cloud computing in 5g. *IEEE Network*, 29(2):32–38.

Constantinides, M., Dowell, J., Johnson, D., and Malacria, S. (2015). Exploring mobile news reading interactions for news app personalisation. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pages 457–462.

Curumsing, M. K., Fernando, N., Abdelrazek, M., Vasa, R., Mouzakis, K., and Grundy, J. (2019). Emotion-oriented requirements engineering: A case study in developing a smart home system for the elderly. *Journal of Systems and Software*, 147:215–229.

Curumsing, M. K., Lopez-Lorca, A., Miller, T., Sterling, L., and Vasa, R. (2015). Viewpoint modelling with emotions: a case study. *International Journal of People-Oriented Programming (IJPOP)*, 4(2):25–53.

de Lange, P., Nicolaescu, P., Rosenstengel, M., and Klamma, R. (2020). Collaborative wireframing for model-driven web engineering. pages 373–388.

El-Attar, M., Luqman, H., Karpati, P., Sindre, G., and Opdahl, A. L. (2015). Extending the uml statecharts notation to model security aspects. *IEEE Transactions on Software Engineering*, 41(7):661–690.

Goncalves, E., Castro, J., Araujo, J., and Heineck, T. (2018). A systematic literature review of istar extensions. *Journal of Systems and Software*, 137:1–33.

Grundy, J., Khalajzadeh, H., and Mcintosh, J. (2020). Towards human-centric model-driven software engineering. pages 229–238.

Grundy, J. and Patel, R. (2001). Developing software components with the uml, enterprise java beans and aspects. *Proceedings 2001 Australian Software Engineering Conference*, pages 127–136.

Holzinger, A., Searle, G., Kleinberger, T., Seffah, A., and Javahery, H. (2008). Investigating usability metrics for the design and development of applications for the elderly. In *International Conference on Computers for Handicapped Persons*, pages 98–105. Springer.

Hussain, A., Abd Razak, M. N. F., Mkpojiogu, E. O., and Hamdi, M. M. F. (2017). Ux evaluation of video streaming application with teenage users. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 9(2-11):129–131.

Kamalrudin, M., Grundy, J., and Hosking, J. (2012). Maramaai: tool support for capturing and managing consistency of multi-lingual requirements. pages 326–329.

Kulyk, O., Kosara, R., Urquiza, J., and Wassink, I. (2007). Human-centered aspects. In *Human-centered visualization environments*, pages 13–75. Springer.

M. Roser, E. O. O. and Ritchie, H. (2019). Life expectancy. *https://ourworldindata.org/life-expectancy*.

Masood, M. and Thigambaram, M. (2015). The usability of mobile applications for pre-schoolers. *Procedia-Social and Behavioral Sciences*, 197:1818–1826.

Michaels, G. M. (1924). Colour preference according to age. *The American Journal of Psychology*, 35:79–87.

Pan, X. (2010). Research of iphone application ui design based on children cognition feature. In *2010 IEEE 11th International Conference on Computer-Aided Industrial Design & Conceptual Design 1*, volume 1, pages 293–296. IEEE.

Parker Software (2019). Key considerations for making age-friendly software. *https://www.parkersoftware.com/blog/key-considerations-for-making-age-friendly-software/*.

Pew Research Center: Internet, Science & Tec (2019). Internet/broadband fact sheet. *https://www.pewresearch.org/internet/fact-sheet/internet-broadband/*.

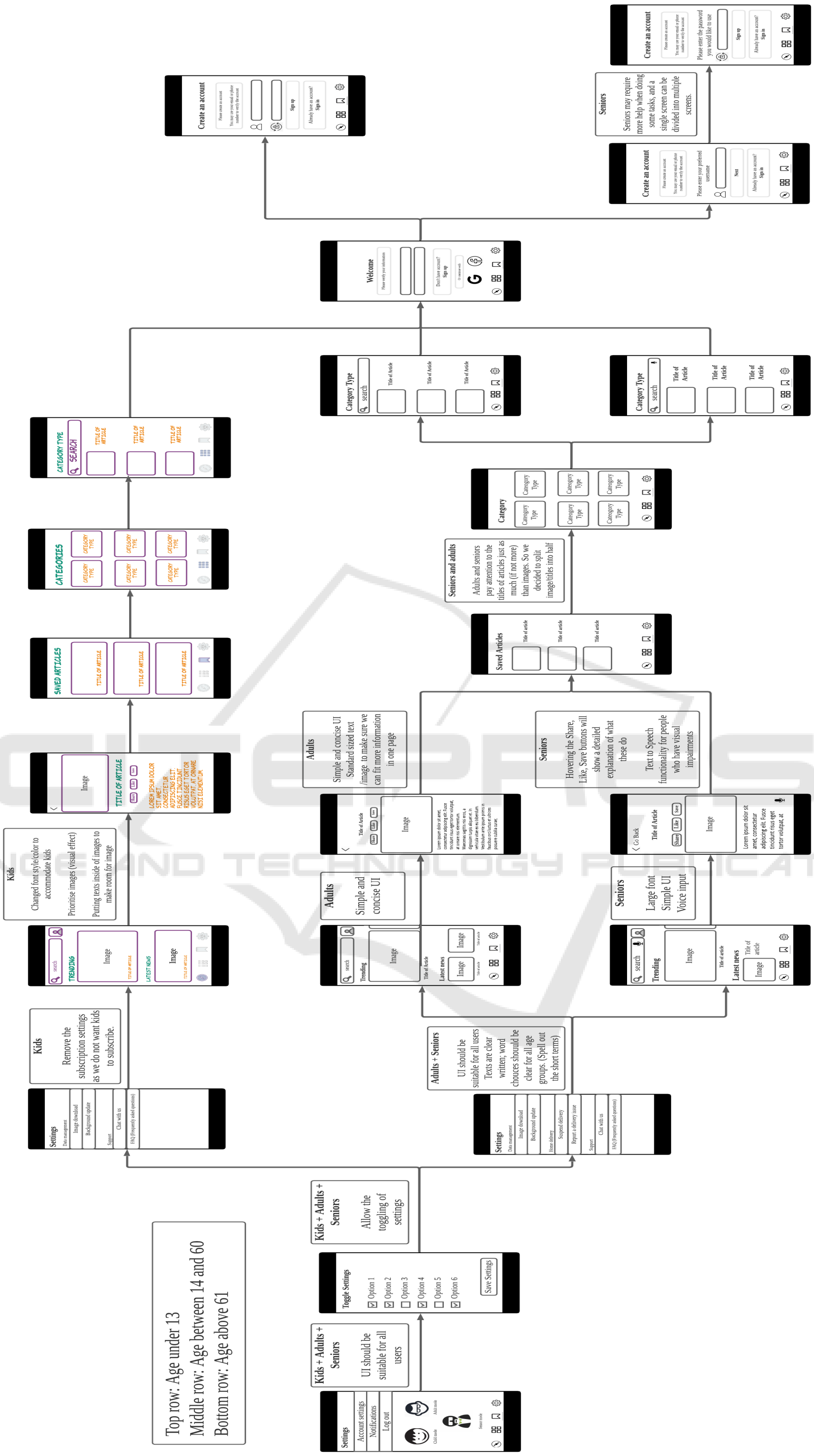
Spichkova, M., Zamansky, A., and Farchi, E. (2015). Towards a human-centred approach in modelling and testing of cyber-physical systems. pages 847–851.

Statistic Canada (2017). Age categories, life cycle groupings. *https://bit.ly/3dopioH*.

Wagner, N., Hassanein, K., and Head, M. (2014). The impact of age on website usability. *Computers in Human Behavior*, 37:270–282.

APPENDIX

The full extended wireframe designed to create the news app prototype is shown in figure 13.



Top row: Age under 13
Middle row: Age between 14 and 60
Bottom row: Age above 61

Kids + Adults + Seniors
UI should be suitable for all users

Kids + Adults + Seniors
Allow the toggling of settings

Kids
Remove the subscription settings as we do not want kids to subscribe.

Adults + Seniors
UI should be suitable for all users Texts are clear written; word chouces should be clear for all age groups. (Spell out the short terms)

Kids
Changed font style/color to accommodate kids
Prioritise images (visual effect)
Putting texts inside of images to make room for image

Adults
Simple and concise UI

Seniors
Large font
Simple UI
Voice input

Adults
Simple and concise UI Standard sized text /image to make sure we can fit more information in one page

Seniors
Hovering the Share, Like, Save buttons will show a detailed explanation of what these do
Text to Speech functionality for people who have visual impairments

Seniors and adults
Adults and seniors pay attention to the titles of articles just as much (if not more) than images. So we decided to split image/titles into half

Seniors
Seniors may require more help when doing some tasks, and a single screen can be divided into multiple screens.

Figure 13: Extended wireframe designed to create the news app prototype.