

Abstract

louis is an affordable, interactive and assistive device for learning braille. It includes an application ecosystem and has a unique, physically-extensible, modular design. First, we will develop a Lego prototype of the main controller and one refreshable character module that can display a single letter. We then extend this model to include the full braille character set. We will develop the firmware API and a primary ‘main’ application, as well as a teaching flashcard app. Our next step is to incrementally miniaturize our hardware by replacing parts with smaller, 3D-printed, alternatives. We will develop two additional character modules to showcase our modular design. And finally, we will add audio input support, enabling more advanced apps to be developed and added to the application ecosystem.

1. Goal description

Whether it exists on public signage, translated books, or on important medication, braille literacy is an invaluable skill for visually impaired people which gives a freedom in everyday life that sighted people so casually take for granted. Whilst many visually impaired people use braille only for more practical reasons, it is also valuable in a variety of situations; Former Home Secretary David Blunkett in 2012 urged people to learn braille, arguing that electronic displays and audio were still unsuitable substitutes for braille in several situations, such as delivering presentations (Rose, 2012).

The problem we aim to solve is decreasing braille literacy rates among blind or visually impaired people. At a previous estimate fewer than 1% of visually impaired people in the UK were users of braille (Rose, 2012). Our solution is an interactive refreshable braille robot, coupled with fun and approachable software which can help both children and adults learn the universally accepted and useful system of braille.

1.1. Relevance of the system

Since *louis* primarily serves as an aid to learn and interact with braille, our product will be most relevant to visually impaired people and their support network (parents, teachers, etc.).

1.1.1. THE IMPORTANCE OF BRAILLE

It has been argued that braille is becoming a less relevant tool, given digital technology’s prevalence throughout our society. However, studies have shown that students who can read braille tend to acquire higher literacy rates (Ryles, 2004). One survey conducted by Louisiana Tech University

found that people who learn braille have a much higher chance of securing a job (T. Alexander, 2012). Braille education is crucial to literacy, and literacy is crucial to employment.

Braille allows users to learn skills like spelling, punctuation and capitalisation, whereas a common alternative, audio, does not. Audiobooks have provided an excellent additional resource for reading comprehension, but passive listening does not have the same effect as active reading and writing. Audio transcription is also not always appropriate or viable in certain situations, for example in a crowded lift. Braille offers a system for labelling all kinds of items (e.g: lift buttons and medicines) which aids independence and raises self-esteem.

1.1.2. THE CURRENT MARKET

Even though learning braille has been proven to be beneficial, literacy rates are low. In the UK less than 5% of legally blind people have the ability to read braille (A. Phillips, 2011). Among the causes are school budget constraints and personal financial limitations. The prices of the current devices on the market are described by users as “*phenomenally high*” (H. Cryer, 2011), and justly so. Modern notetakers, which include some Android and Windows applications, are sold for more than £800 and learning tools are priced at over \$2,000. Another issue raised in the user research is the necessary repairs, which are “*extremely inconvenient, [...] not to mention the expense*” (H. Cryer, 2011). Our modular architecture means that users won’t have to go without their devices, as they will be able to use as many, or as few braille cells as they need. This also keeps the initial cost of our product low, with ‘pay-for-what-you-use’ characters to grow your braille display.

Moreover, most refreshable braille learning devices that are currently on the market are aimed at children, whereas our product is aimed at users of all ages. It can support all kinds of learning apps, as well as serve as a more general assistant. With our open SDK, our robot can be useful, relevant and accessible to everyone.

1.2. High-level description

louis is primarily a voice controlled learning assistant, complete with a speaker, that helps visually impaired individuals learn braille easily and conveniently. It makes use of quizzes and games to allow its users to learn from basic braille characters to more advanced concepts such as contractions.

A unique aspect of *louis* that distinguishes itself from its competitors is that it has a modular architecture for both software and hardware, therefore the users only need to purchase as many parts as they believe will be helpful. This would give an affordable opportunity for any of those people wanting to learn braille, but are unable to do so due

to financial boundaries. *louis* also provides an application ecosystem where developers can build software using the development toolkit included with the product. Developers could create new tools and applications such as an app to read news headlines, smart speaker integration, a braille music tutor and countless others. The open approach with software development makes our product truly useful beyond our small team.

1.2.1. USER STORIES

1. As a user, I can learn braille by using a flashcard app on the device.
2. As a user with a limited budget, I find *louis* a lot more affordable than its competitors as I only need to purchase parts that I believe will be useful.
3. As a visually impaired user, I can open applications of my choice with my voice.
4. As a developer, I can build applications on the platform *louis* provides using the given development toolkit.

2. Task planning

We have identified our main technical goals to be the following:

1. User selection of letters, display speed and application
2. Start of flashcard or other application
3. Choice of next letters to be displayed by application
4. Output of braille letters on device cells
5. Audio output of letters through speaker

The two main dependencies concern the output of braille letters, as the teaching purpose of the device entirely depends on the braille representation to also be translated and repeated out loud to the user.

Another one concerns user input, in terms of application selection and letter display speeds. The input method can be chosen at a later date, as physical or audio inputs would yield the same results, but this information needs to be present first before the device can generate the next letters for display.

These main system requirements are summarized in the goal model shown in Figure 1.

2.1. Milestones

We have defined the following milestones and evaluation measures based on the technical goals defined in the section above:

2.1.1. HARDWARE

1. Working display mechanism with pins
 - (a) Milestones:
 - i. Motor rotates, pins move up and down
 - ii. Two additional 3D-printed cells
 - iii. Audio or physical input integration
 - (b) Evidence and evaluation:
 - i. Correct display of one specific letter (e.g 'A')
 - ii. Correct display of sequence of three letters
 - iii. Audio output confirming speed and app choice
2. Casing
 - (a) Milestones:
 - i. Prototype in Lego
 - ii. 3D-printed prototype
 - iii. Enclosure and fittings

2.1.2. FIRMWARE AND SOFTWARE

1. Milestones:
 - (a) Core app passes on hardcoded letter to display
 - (b) Physical and audio output of letters and Flashcard app: core app passes on flashcard app letters at given speed
 - (c) App selection and reading speed selection: core app receives either audio or physical selection from user for display speed and app
 - (d) Other app(s): core app receives user app choice and selects correct one
2. Evidence and evaluation:
 - (a) Letter displayed after a reasonable amount of time
 - (b) Physical and audio display of letters at correctly timed intervals (and audio output returned after physical, also timed appropriately)
 - (c) Letters displayed at correct speed for each choice (can time it); correct app selected
 - (d) Correct letter sequence for each additional app is returned

2.2. Task decomposition

The required tasks for the project were split into atomic units. A Gantt chart, shown in figure 2, shows the progression of these atomic units throughout the time available for the task, as well as highlighting the dependencies between the atomic units.

To make the high-level tasks more manageable, we will be routinely incorporating story decomposition into our workflow. The process comes from the agile methodology, as outlined in the workshop for project planning. By breaking down stories into smaller tasks, we will increase iteration speeds and regularly deliver incremental value gains.

2.3. Resource distribution

2.3.1. HUMAN RESOURCES

Our company, ThreeDots, is composed of ten members. Over the next eleven weeks, each will commit approximately 200 hours to our project, *louis*.

Ten percent of this time will be remitted to allow individuals to complete their final report. This is proportional to the allocated marks (relative to the rest of the project), as well as to the time available after the final demonstration (one week).

This leaves 180 hours per capita, (a combined 1,800 hours), to allocate between:

1. Iterating on hardware, firmware and software
2. Preparing for, and presenting at, demonstrations and workshops
3. Creating reports, user guides and other technical documentation
4. Administrating and participating in meetings
5. Performing market research and product evaluation

The allocation of these hours is shown in Table 1.

Intentionally, we have not assigned ten percent of the available time, to account for natural project slippage and to alleviate unforeseen delays.

2.3.2. HARDWARE, FINANCE AND TECHNICIAN TIME BUDGETING

Initially, we will create a proof-of-concept hardware design using the Lego EV3 Mindstorms kits that have been provided, and incrementally replace various parts with 3D-printed pieces. This will allow us to miniaturize our components, and create custom structures. We also plan to utilize the Raspberry Pi micro-computer to act as the main robot controller, and to interface with the audio I/O.

Financial budgeting and the tracking of technician time will be completed using a shared Google Sheets file to ensure transparency within the group, and allow for adequate planning. We anticipate that the majority of both resources will be consumed by 3D-printing these custom parts to miniaturize our hardware. Finally, we hope to enclose our components in a branded acrylic plastic enclosure to help with marketing the robot. To minimize inflicted costs, we will utilize some group members' previous personal experience to create CAD designs, and export to STL models for printing, engraving etc.

At this time, we do not anticipate requiring any custom, externally-ordered components.

The simplified bill of materials and budget estimation calculations have been provided in Table 2.

Task	Hours
Hardware	
First iteration	87
Second iteration	116
Third iteration	87
Enclosure and fittings	95
Firmware & Software	
Firmware	90
Central application	141
Additional applications	345
First demonstration	
Report	40
Meeting	10
Second demonstration	
Report	40
Meeting	10
Third demonstration	
Report	36
Meeting	10
User Guide	60
Trade Fair	
Report	50
Meeting	80
Support	
Workshops	20
Q&A sessions and office hours	20
Meetings	
Scheduled (with mentor)	120
Additional (impromptu)	140
Market Research	
Ethics	8
Initial contact	5
First session	5
Second session	5
Total	1620

Table 1. Human Resource Allocation

Item	Technician Cost (hours)	Financial Cost (£)
Lego EV3 Mindstorms Kit	-	-
Raspberry Pi	-	-
3D-printed Components	4.5	40
Wiring	-	8
Acrylic enclosure	4	30
Miscellaneous	1.5	20
Total Cost	10	98

Table 2. Simplified Bill of Materials and Budget Estimation

2.4. Risk assessment

Risks regarding the project and its development were measured with respect to Table 3 and Table 4.

Scale of Risk	Severity of Risk
Low (1)	Little to no impact
Medium (2)	Would affect the progress of the project, but it can be contained
High (3)	Major setback

Table 3. Severity of Risk

Scale of Risk	Likelihood of Risk
Low (1)	Unlikely, but possible
Medium (2)	May occur, but not very likely
High (3)	Risk is likely to occur

Table 4. Likelihood of Risk

Potential risks identified and corresponding actions that will be taken if we are required to face them are described in Table 5.

2.4.1. STRENGTHS, WEAKNESSES, OPPORTUNITIES AND THREATS

The strengths, weaknesses, opportunities and threats of this project have been outlined in Table 6.

3. Group organisation

The organisational structure is based on the distribution of pre-existing skills as to increase the probability of on-time delivery of the project while still accommodating individual team members' desire to learn new skills in areas they lack expertise. The project manager is responsible for activity and resource allocation, as well as risk analysis and contingency planning, whereas the secretary is concerned with the management of deadlines and meeting planning.

The rest of the team is divided into the functional units of *hardware* and *software*, and sub-units for system design, and low-level/high-level for software. Communication between the system design role of hardware and software is essential in order to ensure the interoperability of the systems developed independently by the team members.

All communication is handled on Slack to avoid scattering of information and to ensure consistency and transparency. The number of channels is reduced to a minimum.

Fixed weekly meetings for the whole team are scheduled in advance to ensure the availability of all team members to discuss the current progress of the project and potentially plan adjustment measures. Additionally, team members are committed to daily stand-ups on Slack with the aim of synchronizing information among the team and identifying potential issues.

Code-sharing and version control are all handled using git and GitHub. In order to guarantee the cleanliness and

functionality of the code base, a contribution procedure is enforced by which commits cannot be pushed directly to the master branch but must be submitted using pull requests. The pull requests are then peer-reviewed by other team members and automatically checked for compilation errors in the continuous integration pipeline.

A graph detailing the group subdivision by functional unit is included in Figure 3.

References

- A. Phillips, L. Beesley. Braille profiling project. Technical report, Royal National Institute of Blind People, 2011. URL https://www.rnib.org.uk/sites/default/files/braille_profiling.doc.
- H. Cryer, S. Home. Use of braille displays. Technical report, Royal National Institute of Blind People, 2011. URL https://www.rnib.org.uk/sites/default/files/2011_06_Use_of_braille_displays.doc.
- Rose, D. Braille is spreading but who's using it?, 2012. URL <https://www.bbc.com/news/magazine-16984742>.
- Ryles, R. Research study: Early braille education vital. *Future Reflections*, Special Issue, 2004. URL <https://www.nfb.org/images/nfb/publications/fr/fr14/fr04se22.htm>.
- T. Alexander, S. King, S. Rothstein. Transforming braille project charter. Technical report, The Transforming Braille Group LLC, 2012. URL <https://www.transformingbraille.org/wp-content/uploads/2014/08/Transforming-Braille-Project-DAISY-Charter-v8-25-June-2012.doc>.

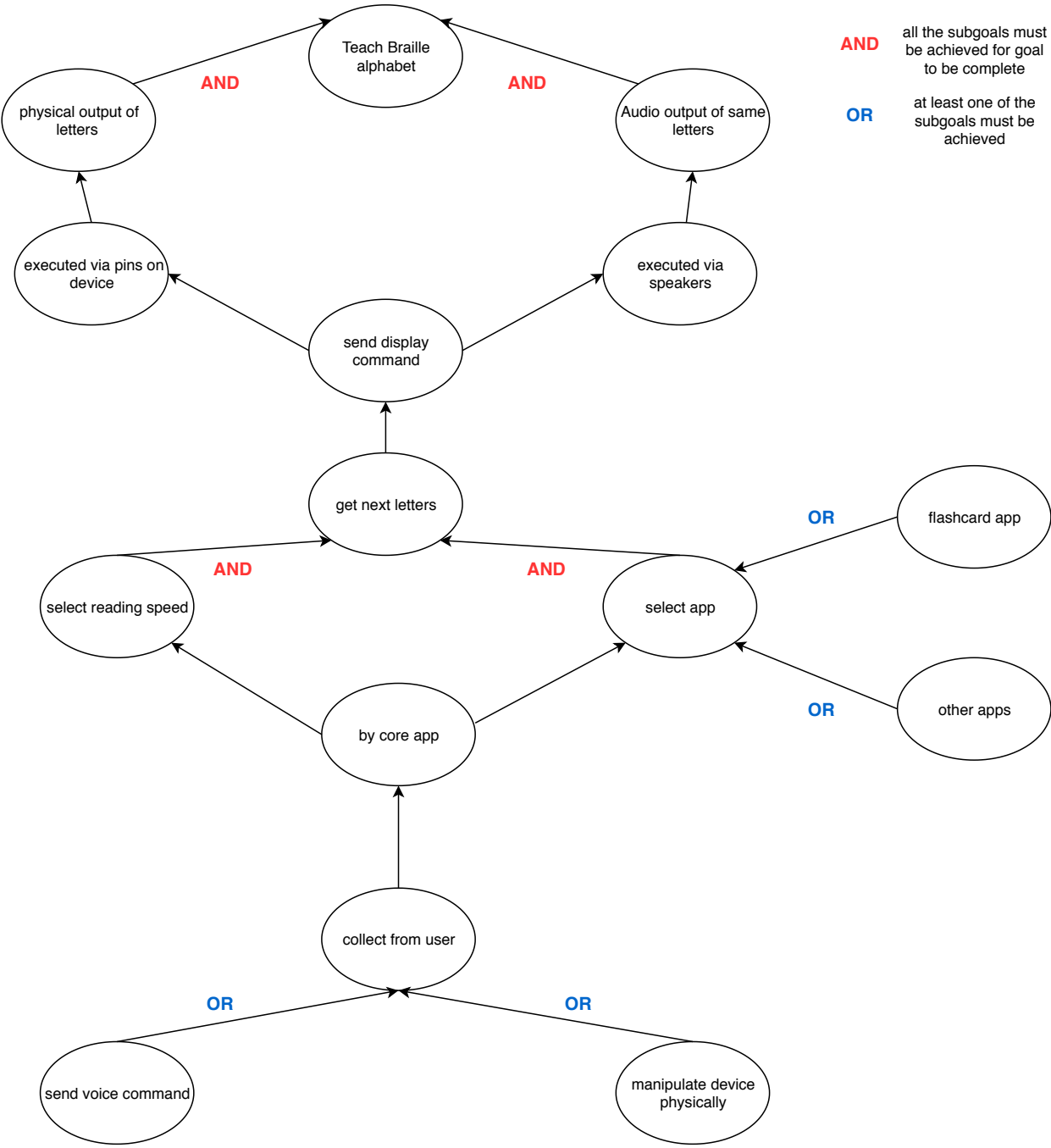


Figure 1. System Requirements Goal Model

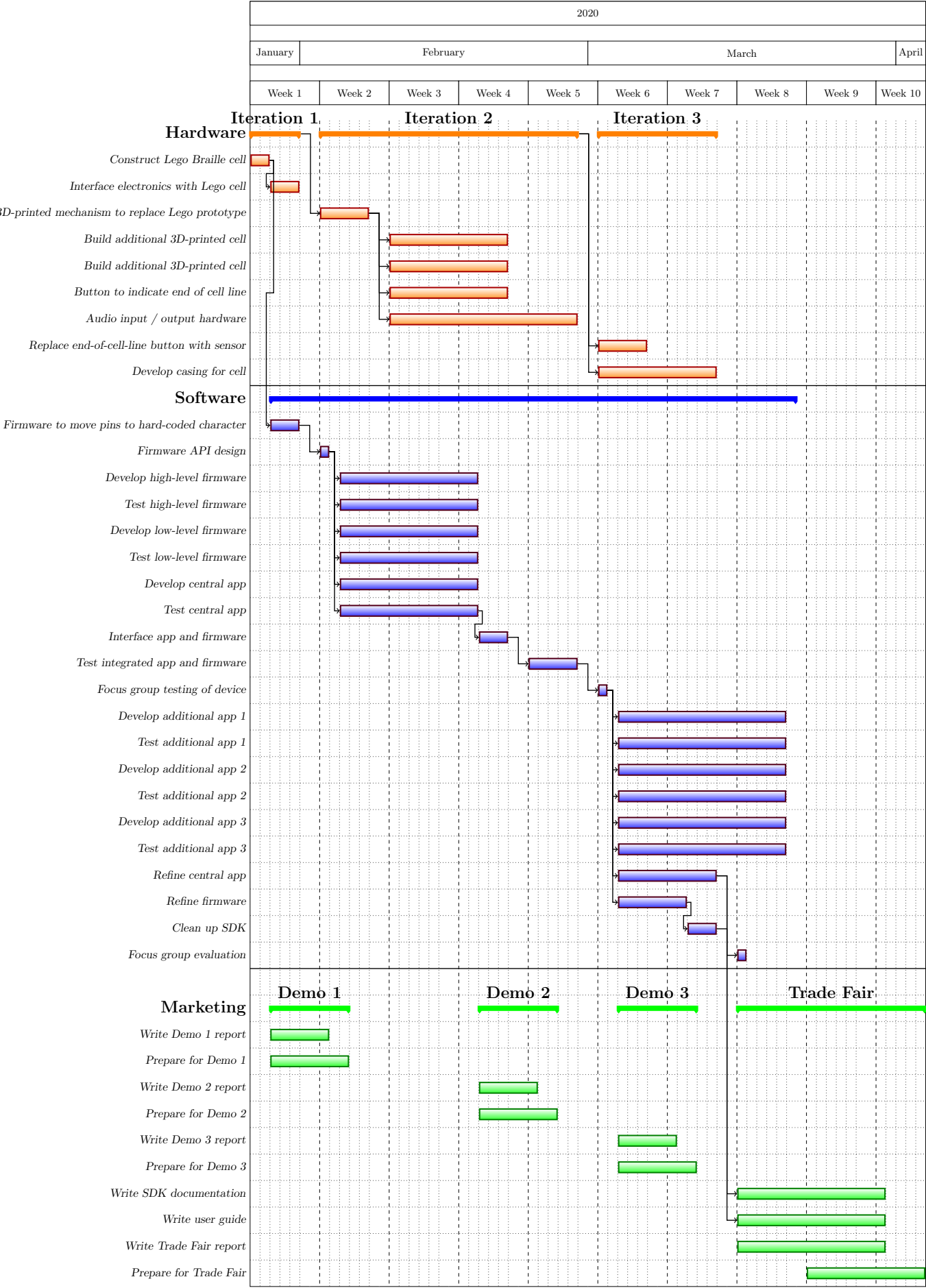


Figure 2. Gantt chart of task decomposition and dependencies

Risk	Impact on project	Measures taken to minimize	Severity	Likelihood	Contingency plan
Incorrect translation to braille from alphabet	Device works fine, but becomes not usable	We will be studying the direct translation and make sure that each letter is translated correctly	3	1	Allow for testing time, to check that machine works as intended
Team member becomes unavailable due to emergency and/or absence	Work falls off schedule or is submitted but incomplete	Multiple group members are assigned to each team, so that work is divided and easily taken over if needed	3	2	Team members with less workload should switch teams if needed
Hardware falls apart	Device is not able to work. Will lose time on fixing and retesting the device	Allocated 10% of our resources in time and budget for potential mishaps and this is included	3	2	Finish with the hardware as early as possible, to allow for any problems to be spotted as well as avoid any last minute risk
Clashes between team members ideas/working hours/working styles	Lose communication between members, team is not focused on tasks and there could be delays in completing them	Ideas are run through the team and are evaluated in terms of feasibility and effectiveness. Times and style are discussed between team members from the beginning and are followed so as to avoid clashes	3	2	If a clash is not resolved within the team, then the issue is raised in the entirety of the group and if needed to the Course Organisers
Device is not easy to use	The purpose of the machine is defeated, since most of the users will not be able to use it	Research project and produce a good System Design. Be in cooperation with a visually impaired person suitable to establish if the device is hard to use. Establish modularity of the design	3	1	Allocate testing time to allow for adjustments to be made easily. If we cannot make it easy to use, then maybe rethink the purpose of the device
Delays in building functional hardware results in delays in building the software.	Might result in producing an incomplete application with regards to software	Once the first prototype of the braille cell mechanism is built, software and hardware development can be done in parallel	3	1	Allocated 10% of our time in potential delays and errors
Problem in building the original prototype of the braille cell mechanism entirely from lego parts	Delays in building the hardware, as it is the main aim for the first demo	Design the system in detail and adapt to any adjustments needed as an initial compromise and improve in 2nd iteration	3	1	3D print the required parts and complete the lego structure
Not able to create each braille cell as small as possible	Device is considerably bigger than initially estimated and thus more material is used; bigger budget	Thorough system design including hardware and software, as well as budget allocation	3	1	Adjust our goal, so that we are more flexible regarding the size of the cell
Not successful in creating several quizzes and games	Some of the purposes of the device are not developed	Ensure we build applications that are easy; correct study and design of software	3	2	Adjust our goal; maybe focus on building one game to present as a prototype

Table 5. Identified risks and corresponding actions

Strengths (internal factors)	Weaknesses (internal factors)
<ol style="list-style-type: none"> 1. Experience and Skill levels (hardware designing, low and high-level programming, and marketing) 2. Resource availability (Lego, controllers, and 3D printer) 3. Unique solution - no other product alike on market 4. Effective leadership and planning 5. Clearly defined goals 	<ol style="list-style-type: none"> 1. Gaps in knowledge and expertise 2. Unforeseen technical challenges (both software and hardware) 3. Other members' commitments
Opportunities (external factors)	Threats (external factors)
<ol style="list-style-type: none"> 1. Technology and infrastructure development 2. Market demand 3. Availability of potential testers of our product 	<ol style="list-style-type: none"> 1. Timescale and deadlines 2. Other competitors' solutions on market 3. Limited budget

Table 6. Strengths, weaknesses, opportunitites and risks

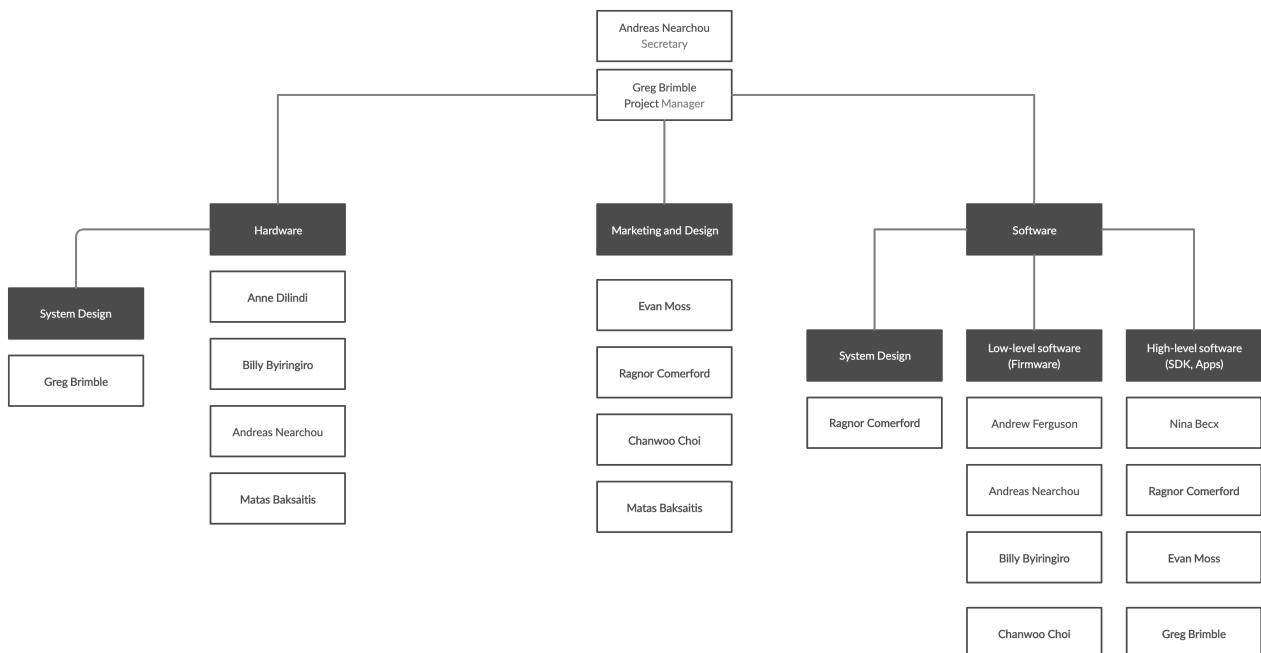


Figure 3. Group member organisation and subdivision by functional unit