Bilkent University
Department of Computer Engineering

CS 491- Senior Design Project

Papyrus

Project Name: Papyrus

High-Level Design Report

Group Members: Ant Duru, Atay Kaylar, Enver Yiğitler
Supervisor: Fazlı Can

Jury Members: Shervin Arashloo, Hamdi Dibeklioğlu

Innovation Expert: Murat Kalender

1.Introduction

Learning or mastering a new language is an important instrument for enhancing one's personal growth and has a number of pragmatic uses in intrinsic and extrinsic domains. It improves job opportunities, sets the stage for obtaining more complex relationships with foreign people or even providing a deeper understanding of self in a person[1]. One of the greatest authors Johann Wolfgang von Goethe would say "He who knows no foreign languages knows nothing of his own.", where studying language opens the door for deeper understanding of various cultures through its embeddedness into a nation's collective unconscious. With understanding of others, one can comprehend itself more profoundly. Hence we as a team knew the importance of language learning and opted to create an application that helps people to further train on a specific language. *Duolingo* is the most popular language learning application in the market[2], it helps people to learn new languages with a variety of game-like exercises. When it comes to learning words or fundamental structures, it is very useful, however it is not enough for mastery[3]. Kató Lomb is a Hungarian multilingual that has self-taught herself 16 languages. In order for her to master or even start to learn new languages, she insisted on learning the context of sentences, thereby the complex expressions, or the unknown words were redundant for her[4]. Papyrus provides a word learning tool, where the user defines flashcards, however we acknowledge the fact that learning a language comes from reading and understanding the context just like Lomb addressed, and we constructed our application according to it.

1.1 Purpose of the System

Papyrus is a mobile app that submits varied language learning and training tools and methods with the help of machine learning processes in a social-media like environment. Kató Lomb proposed that language mastery essentially comes from understanding a context without the details of convoluted expressions and it is redundant to check the dictionary every time an unknown word pops up. However, still we do think that one should have a rich vocabulary of words and one should train on the unknown words on a regular basis. Also we acknowledge that one's understanding of language greatly enhances when reading or hearing a book in a comprehended context. These arguments constitute the backbone of ways we intend our application will help people for language learning. In that way we aim to craft the optimal language setting for the end-user, with use of flashcards that exercises unfamiliar words on users, and *machine-learning* enhanced reading environment, where users can simultaneously read a book in selected multiple languages, for effective context understanding. Moreover we intend to implement audiobook and narration options into books, with the alignment of audios and texts. *Papyrus* has a number of functionalities other than previously explained ones, following are four headings for clearer explanation of our project's major features.

- **Sentence Aligned Text Reading**

  If available, books will be matched with their versions in other language sentence to sentence. Thus, users will be able to see the translated version of the sentence they do not know in the language they want to learn, whenever they want.

- **Narration, audiobook, text, all in one**

  The app also includes a library of audiobooks aligned with the texts, sentence to sentence. The user can hear the narrated version of the sentences in the book at any time by selecting the sentences they want.

- **Annotations**

  Users can highlight selected text parts, and note on the highlight.

- **Flashcards for advanced word memorization**

  Users will be able to create flashcards for desired words, and rate each flashcard good, bad and neutral. In this way, unmemorized words will again be asked to the user. Moreover, user's will be able to download flashcard decks online and study on them.

- **Social media-like environment for compelling app experience**

  Users can create book playlists called *shelves,* write reviews on books, follow authors or other users, and share their annotations. Thus, other users can see other users, reviews, shelves and annotations.

1.2 Design Goals

Usability

- The application should have an easy-to-use interface.

High Performance

- Shifts between menus should be instantaneous.
- Users should be able to turn pages of the books without any noticeable delay.

Reliability

- The alignments of the sentences should be reliable.
- The instantaneous translations should be reliable.

Marketability

- The application should attract customers in order for publishers to share their books with the application.

Extendibility

- The application will be an iOS application at first, but can be extended to more operating systems for both mobile and desktop.

Security

- No data taken from the users will be shared with third parties.

Scalability

- Large data of users should be kept without any data loss or any other problem.
- The data of the books should be kept without any problem.

Maintabilty

- With opaque layering, each subsystem can be more easily adjusted to the changing environments, and the system can be maintained and adjusted with software updates.

Flexibility

- Client/server model supports a number of devices, in that way the application, just it is scalable, it can be updated for different software environments.

Modularity

- Each subsystem is partitioned for easier implementation and easier designing process. In this way complex application is split into more manageable parts, therefore reducing the complexity

1.3 Definitions, Acronyms, Abbreviations

Term	**Definition, Acronyms and Abbreviations**
Redis	Redis is a data structure server. It is an open source, memory-driven, key-value store. Redis stands for "Remote Dictionary Server". It is the most used key-value database according to various sources. Since June 2015, its development has been supported by the Redis Labs company.[5]

Swift	Swift is a powerful and easy-to-use object-oriented programming language created by Apple to develop iOS and Mac apps for iOS and macOS platforms. It was first announced at the WWDC 2014 conference.[6]
JavaScript	JavaScript is a dynamic programming language that is widely used in web browsers.[7]
Machine Learning	Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.[8]
Database	Databases are areas where structured information or data is stored.
SQLite	SQLite is a transactional and relational SQL database engine, the most widely distributed and recommended source code in the world, publicly available, completely developed with C/C++ programming languages, without server software and configuration requirements. SQLite can be used easily in

	dozens of programming languages.[9]
PostgreSQL	PostgreSQL or Postgres is a free and open source, SQL supported relational database management system.[10]
JSON	JSON; It is a text format that facilitates the exchange of structured data between all programming languages. It's useful in many contexts and applications, with its brackets, square brackets, colons, and commas.[11]
EPUB	EPUB, also known as Electronic Publishing, is an advanced file application in close family and computers, received internationally by the e-books forum.[12]
WebPUB	WebPub is a web interface that allows users to install, import, manage, and upgrade their websites. Its aim is to ensure website security by continually upgrading scripts and keeping client information secure.[13]
AudioBook	The term is used for books, that is in the audio format.
Sentence Alignment	Sentence alignment is the problem of making explicit the relations that exist between

	the sentences of two texts that are known to be mutual translations.[14]
DNS	DNS(Domain name system) is a distributed naming system for any purpose, service or special purpose, to partition the internet space, to organize the boundaries of the region and from the headquarters.
Load Balancer	Load balancing; the technology of sharing work between two or more computers, processors, hard disks, servers, or other resources.
HTTP	Hypertext transfer protocol: is used for transferring hypertext documents on the internet
BLOB	Binary large object, it is used for big data files, such as AudioBooks.
iOS	iPhone Operating System is the operating system embedded in Apple mobile devices.
KVKK	KVKK(Kişisel verilerin korunma kanunu) is the law of Turkey, which protects personal data in the cyber environment.

Table 1. Explanations for the acronyms and definitions of terms related with the project.

1.4 Overview

This report will be informing about the high-level design of the project. The subsystems will be visualized using UML notations, its backend and frontend services and packages will be explained in detail. Moreover, consideration in engineering design will be analyzed in each factor. The teamwork details and work distribution will be reported. In short the report is essentially about explaining the architecture of the application. The main components of the architecture are server(s), database systems and the mobile device that the application runs on. All three of the components tasks and their interlinked connections will be explained. The architecture is in a 3-layered client/server model, with *opaque layering*, where each layer can communicate with the layer below.

2.Current Software Architecture

Kindle

- Provides narrated text
- Provides dictionaries, machine-translation and Wikipedia search.
- Provides an underdeveloped flashcard system where the user can only write a word on two sides.

duoBooks

- Includes several aligned books.
- Provides translation of a word but without any dictionary or Wikipedia search.
- An undeveloped flashcard system where the user can only keep a word with its translation.

Parallel Books

- Limited number of aligned books.
- Can only use Apple’s dictionary.

3.Proposed Software Architecture

3.1 Overview

Papyrus uses client/server architecture in a 3-layer way. The application uses servers and databases that enable client’s usages. The architecture of server and database is simply explained in figure 1. Furthermore, in figure 2 subsystem decomposition of the presentation, application logic and the data layers are shown with their included packages. The environment of the application will be iOS, in this way we will be using Swift as the main programming language, and JavaScript will be used for script base operations in the application and for the interface coding.

3.2 Subsystem decomposition

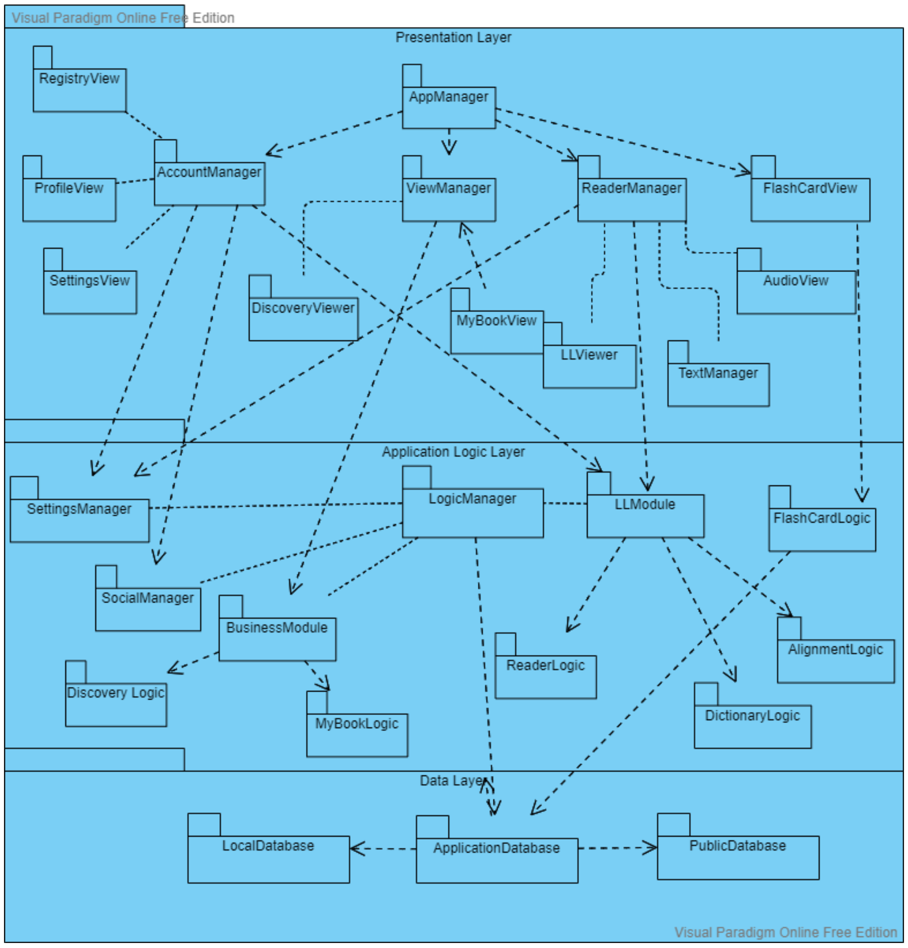

Figure 1. 3-layer architecture consisting of presentation, logic and data layer with their packages

Our application consists of three distinct layers, which are *presentation layer, application logic layer and data layer*. Each layer only communicates with the layer below, therefore we use *opaque* layering, with it, each subsystem can be more easily adjusted to the changing environments, and the system can be maintained and adjusted with software updates. With 3-layer architecture, we aimed for *modularity,* in this way meant to reduce the complexity of the program. The presentation is responsible for displaying the visuals of the application, due to high functionality of the application, this layer has a complex structure. The application logic layer is responsible for the logic and the computational operations for the program, underneath the visuals. Third layer is the data layer that is used for data management. The data for the program comes from either the local repository or the online database. Following sections will explain each layer. All packages will be explained in detail in *section 4.*

3.2.1 Presentation layer

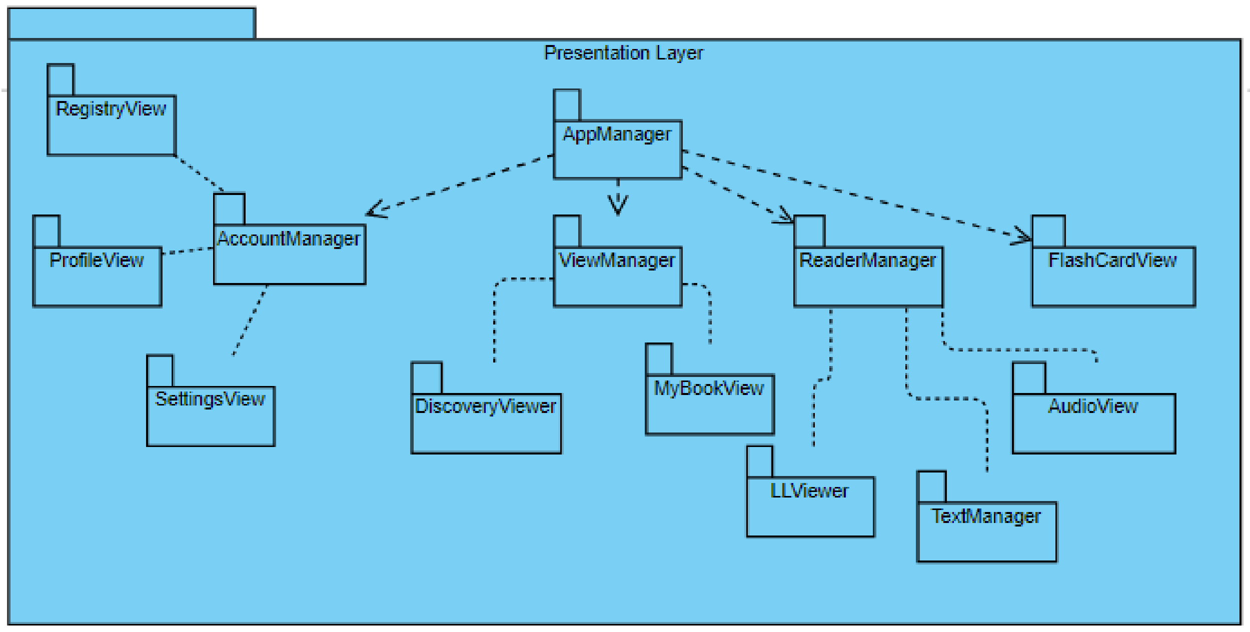

Figure 2. Presentation layer

This layer is responsible for the management of the visual information and displaying the graphics, as well as the auditory material. The main package in this layer is *AppManager* that handles the interactions with other packages. App manager *uses AccountManager*, *ViewManager*, *AccountManager*, *ReaderManager*, *FlashCardView,* which are connected to the viewer package. The viewer packages are lower level packages that are used for more specific visual information processing. *Managers* handle the interactions with *viewers* that are connected to the parent *managers*. Figure 2 shows the relation between the logic and the presentation layer, it is explained in detail in *section 3.2.2.2*.

3.2.2 Application logic layer

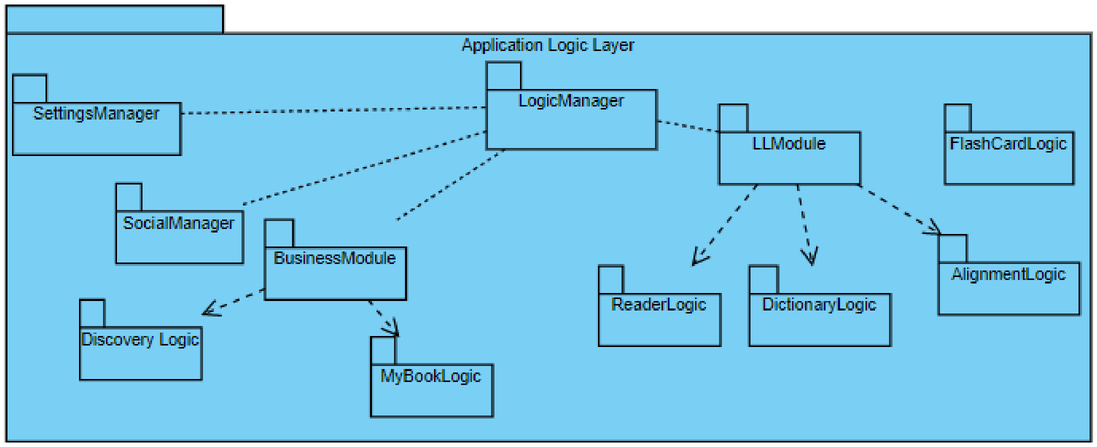

Figure 3. Logic layer

Logic layer is in charge of the logic behind the visuals. Presentation layer uses the logic layer. *LLModule* stands for language learning model which does the language learning features of the application it is used by *ReaderManager*, which views the visuals for language learning and also for review or annotations features of the application.Those features are viewed through *AccountManager*. *FlashCardLogic* is in charge of the algorithms and the operations behind the FlashCard features, it is used by *FlashCardView*. *SocialManager* is responsible for social media features' logic of the application, which is used by *AccountManager*(which views profiles).

3.2.3 Data layer

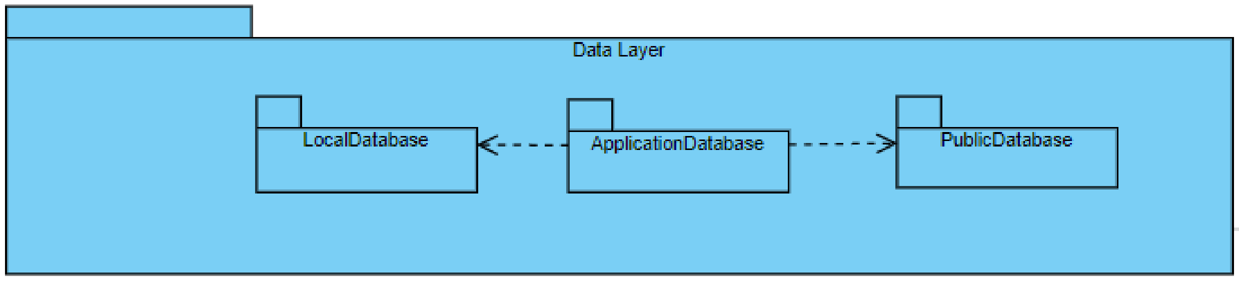

Figure 4. Data layer

Data layer controls data tasks of the application. *ApplicationDatabase* is the main package that is responsible for it. It uses local and online databases through *LocalDatabase* and *PublicDatabase* respectively. *ApplicationDatabase* is used by *LogicManager*, and the desired data is communicated through other connected packages. It is also directly used by *FlashCardLogic*, which uses the desired information about flashcards.

3.3 Hardware/software mapping

Essentially our project will live in three hardware components, the mobile device that runs the application, host that runs the database and the server. The database will use *Redis* for storing cache information, and *PostgreSQL* for storing the chunk of the data. The application will also access information through *SQLite* in the local repository. The application will communicate the online database through a Unix Host server. The server will use the database to handle user operations. The device and the server will communicate through HTTP protocol. The backend will return HTTP to JSON requests coming from the application.

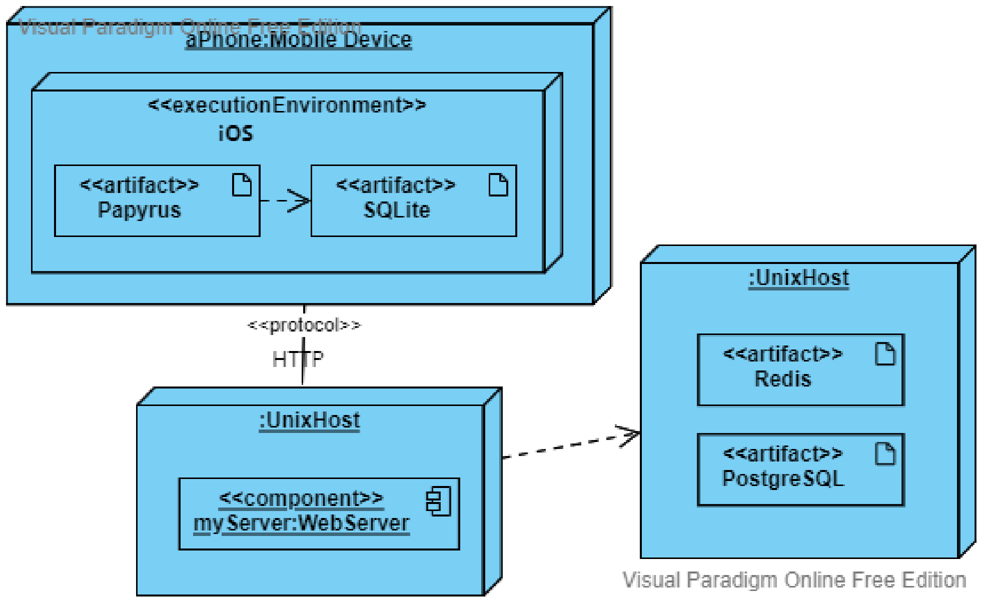

Figure 5. Deployment diagram, that shows the interaction between hardware and software

3.4 Persistent data management

Redis is used for in-memory databases, and also for caching. PostgreSQL is used for relational databases and to store most chunks of database and information in the server database. Therefore, we can make sure that our data's persistence. For persistent data management in the local, meta-information, settings, configurations will be held in SQLite, and local books and AudioBooks will be held in the mobile file system, because we didn't want to record big BLOB objects in the database, like long AudioBooks.

3.5 Access control and security

User's identification and confirmation information will not be shared with any third parties. Also, administrators will not have direct access to this information as well. We will obey KVKK. Users will not have access to each other's private information.

3.6 Global software control

Our application does not use cameras or any other complex hardware structures to implement in the system. It only needs synchronization between server and mobile devices that the application runs in. Server(s) will *use* relational databases for accessing data considering the services of the application. Also local databases will be used on mobile devices, such as *file systems* and SQLite for caching etc.. Backend will return *JSON* to *HTTP* for requests coming from the application. *HTTP* offers a pipeline of requests and responses, and reduces congestion of the network.

3.7 Boundary conditions

Initialization

Users should sign in to the *Papyrus* application to enter the system. They should sign up and create an account if they do not have any.

Termination

In order to terminate the application, users can either sign out from the application or close the application.

Failure

Users may encounter errors during signing up and signing in processes due to a problem with the database or the server. Users, also, may get errors while trying to publish their annotation etc. online or downloading existing elements. The reason for these errors may be internet connection problems. For all these errors, a pop-up page will be displayed. Also, users cannot create an account with an email already in the system.

4. Subsystem services

Papyrus uses web servers and databases for its operations, therefore it is client/server architecture. Firstly it uses DNS servers, or local cache to communicate with our server(s). Load balancers are used for the data trafficking between user devices and web server(s). As it can be seen, application services access the desired data through the database system. Services logic will be done in web servers, and the desired information, will be fetched and communicated through the database. The main services that will be communicated through server and databases are market, search, social and registry/authentication. These processes are visualized in *figure 6* below.

Market service

This service is responsible for creating the market environment for the users, where they can search their desired books and AudioBooks.

Search service

It can be called a search engine for the application, where it shows the desired items, when it is searched by the users.

Social service

This service is responsible for social services of the application, where users can follow each other or authors.

Registry/Authentication

This service is responsible for registration and authentication of the accounts for the application. (More detailed explanation of how this services is run is explained in detail is in *section 4.2)*

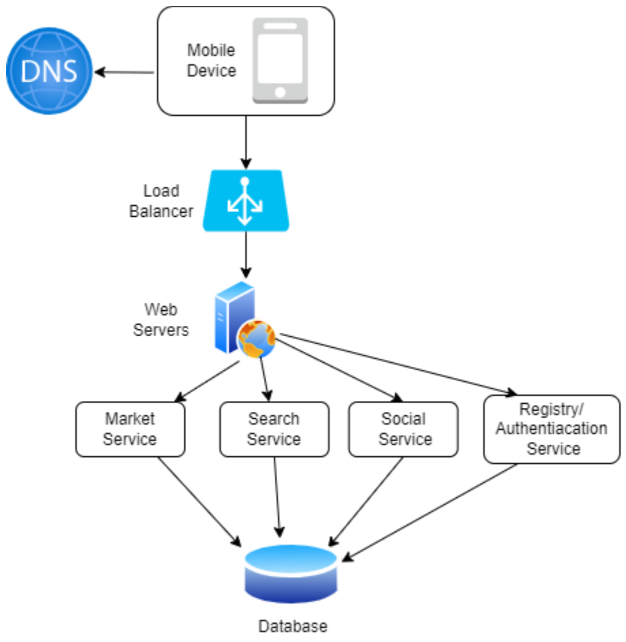

Figure 6 Overall structure of the client/server nature of the application

4.1 Components of the hardware

<<executionEnvironment>>

In CS491 and CS492 we aim to develop the project in iOS platform, in that way the environment, which the application will run is iOS

<u>myServer:WebServer</u>

Server part will be handled by a UnixHost and will *use* the database systems for executing the application on a global scale, computations will be done here. It will be connected to the device using HTTP protocol for fetching hypertext documents, and for easier displacement of the visuals.

PostgreSQL

It will be used for relational database for storing the application data, the big storage of the application

SQLite

SQLite will run on local for local files, it will be run on the execution environment.

Redis

Redis will be used for caching in the database system.

4.2 Presentation Layer

Following packages are used for presenting the visuals for the application.

AppManager

It is responsible for the highest level construction of the view of application. It uses ViewManager, AccountManager, ReaderManager and FlashCardView.

AccountManager

Account manager communicates between RegistryView, SettingsView and profile view to visualize the account informations(such as account page, account settings)

RegistryView

This view will visualize registry and login pages for the application

ProfileView

This view will visualize profile page, and its subpages

SettingsView

This package will be used for presenting the settings pages.

ViewManager

This package is the main package for discovery page and the user’s book pages and will communicate between DiscoveryView and MyBookView

DiscoveryViewer

This package will view the discovery page

MyBookView

This package will view the user’s books page

ReaderManager

This package is responsible for showing reader material, AudioBook and books. It gets it setting information and logic from *SettingsManager*.

LLViewer

LL stands for language *learning learning,* which is used for viewing language learning features of application, such as showing alignet sentences, and presenting dictionary

TextManager

This package is responsible for presenting the text of the book

AudioView

This package is responsible for presenting the text of the AudioBook and narration player.

FlashCardView

This view shows the visuals related view flashcards.

4.3 Application Logic Layer

Following packages are used for the logic and the calculations behind the visuals of the application.

LogicManager

This is the main logic that communicates the *SettingManager, SocialManager, BusinessModule, LLModule* and performs the highest level logic operations.

SettingsManager

This manager is behind the settings operations in the application, it is used for numerous packages, for their setting properties.

SocialManager

This manager is for calculating social media features, and holding the logic behind it, it is used by AccountManager for account-based settings. It is used by AccountManager for the relations for viewing profiles based on the social media features, such as showing *followed authors*.

BusinessModule

The main module for communicating DiscoveryLogic, and MyBookLogic

DiscoveryLogic

The package is responsible for finding search results and showing the idle page in the discovery page, when the search button is not pressed(such as showing recommended books).

MyBookLogic

The logic behind the library of the user, it has its own setting classes and sorting classes.

LLModule

Main module behind the language learning properties of the application.

ReaderLogic

Logic behind AudioBook and text-bases will be done here.

DictionaryLogic

Logic behind the dictionary usage will be done here.

AlignmentLogic

Machine learning enhanced sentence alignment algortihm’s calculations will be done here.

4.4 Data Layer

Following packages are used for the fetching and storing data and information in the database.

ApplicationDatabase

The main database of the application, which will use both the server database and the local repository. It is used by a number of packages in the logic layer for data retrieval.

LocalDatabase

Local repository of the application.

PublicDatabase

Server database of the application.

5. Consideration of Various Factors in Engineering Design

The factors in engineering factors in the grand scheme of the *Papyrus'* predicted impacts are analysed and ranked from 1 to 10.

Public Health

One of our intentions of creating *Papyrus* imbue reading and learning language habits to people, which has a number of advantages. Other than intellectual benefits, reading books also has several health benefits to individuals. It decreases age related cognitive diseases, such as Alzheimer's disease, thanks to increased cognitive activation in the brain. Also it reduces stress, which is good for heart related illnesses and helps to alleviate depression symptoms.[15]

Dyslexic people have a hard time reading text based books, in that way our application has the audiobook feature, which is proven to be an easier method for absorbing the book for dyslexic people.[16] Furthermore, our application has audiobook features, so that users can listen and read the book at the same time, to their liking. It is also proven that reading and listening are a great way to study books for dyslexic people.[17]

Public Safety

For the safety of the public, user’s information, such as their credit card information will be confidential and will not be shared with any of the third parties.

Public Welfare

Our model uses a subscription model for user payment plans. Subscription plans are great for keeping users in their monthly budget, because subscription cost is always consistent and predictable, making the application more convenient to access. Users will pay for the subscription to the application. Yet, e-books will be cheaper than hard copy books. Therefore, users will not be affected negatively in terms of economic issues.

Economic Factors

If the business is suitable for the mode, the subscription model can be great for the business owner, in our case publishers and us. Publishers can calculate the predicted revenue more easily, due to the recurrent nature of the subscription model. John Warrilow says, creator of the Value Builder System, “The more guaranteed revenue you can offer a potential acquirer, the more valuable your business is going to be, because a high percentage of the revenue of a subscription-based business is recurring, its value will be up to eight times that of a comparable business with very little recurring revenue.” [18]. Therefore, our intentions of using a subscription based model, is suitable for both the users(the public) and the money earners from the application(us and publishers).

Social Factors

Papyrus aims to create communities of language learners with respect to their favorite authors, genres, books, etc. Furthermore, since people will be able to share their notes and annotations for a book, their book collections, and their flashcard decks; social interaction will be affected positively.

Environmental Factors

According to Cleantech, a single hard-copy book generates about 7.5 kg of carbon dioxide, while this value is doubled for a textbook. However, an iPad generates 130 kg of carbon dioxide in its lifetime. Therefore, having an iPad neutralizes its harm when the user reads approximately 18 e-books.[19]

Consequently, e-books are much more nature-friendly than hard-copy books. By digitalizing the libraries as we aim with PAPYRUS, readers' carbon footprints will also reduce by a significant amount

Cultural Factors

Papyrus aims to broaden the cultural perspective of people by offering the opportunity to learn a new language by reading books of that language. Since literature is a significant field to represent a culture, users will encounter new cultures and gain knowledge of the cultures as well.

Global Factors

Papyrus can be used globally. Language learners from all around the world will be able to enter the application to read books written in a specific language. Also, languages in the application will be updated regularly to present more languages to learn.

Factors	**Effect Level**	**Effects**
Public Health	7/10	Decreasing stress, alleviating depression symptoms. Assisting dyslexic people.
Public Safety	2/10	Confidentiality of user information
Public Welfare	6/10	Subscription model is good for the end-user for users to predict their monthly spending.
Economic Factors	6/10	Subscription model is good for the publisher and us, because monthly revenue is more easily predicted
Social Factors	7/10	Built-in social media features have a way for people to socialize.
Environmental Factors	8/10	Less trees will be cut, because books are electronically published.
Cultural Factors	9/10	Learning new languages has a way of expanding the other cultures' knowledge in individuals.
Global Factors	9/10	Papyrus is a language learning tool that can be used globally.

Table 2. Various engineering factors, their effect and level.

6. Teamwork Details

6.1 Contributing and functioning effectively on the team

Teamwork is essential for constructing a complicated software project, which we profoundly know the importance of effective teamwork distribution. From the start of CS491 we have distributed and brainstormed how we can most logically distribute the work amongst ourselves. While normally groups are made up of five people, our group consists of three people. In that, we lack the manpower compared to five-people groups. However, what we lack in manpower, can be compensated in various ways, which is explained thoroughly in *section 6.2.* The following shows the sections that each group member contributed to the project.

- **Project Specification Report:** We, as a team, brainstormed ideas to come up with the fundamental functionalities of the program as well as think of how we can implement the project, what technologies we will use and what is the potential work distribution.

- **Project Analysis Report**
  - **Introduction, Current Systems:** Enver Yiğitler, Ant Duru
  - **Proposed System:**
    - **Overview:** Enver Yiğitler
    - **Functional, Non-functional, Pseudo Requirements:** Ant Duru
    - **System Models:** Enver Yiğitler, Ant Duru, Atay Kaylar
  - **Other Analysis Elements:** Atay Kaylar

- **High-Level Design Report**
    - **Introduction:** Atay Kaylar
    - **Current Software Architecture:** Atay Kaylar, Ant Duru
    - **Proposed Software Architecture:** Enver Yiğitler, Ant Duru, Atay Kaylar
    - **Subsystem Services:** Enver Yiğitler
    - **Consideration of Various Factors in Engineering Design:** Atay Kaylar
    - **Teamwork Details:** Atay Kaylar

- **Overall Work Distribution in the Implementation**
    - **Frontend:** Enver Yiğitler, Atay Kaylar, Ant Duru
    - **Backend:** Enver Yiğitler, Atay Kaylar, Ant Duru
    - **Machine learning:** Enver Yiğitler, Atay Kaylar, Ant Duru

6.2 Helping creating a collaborative and inclusive environment

As addressed above our group is two people short of the maximum five people team member limit. However, we think that not always less people means less throughput. Cells divide when they become too big, because it gets harder to control the resources. Just like a cell, a concise group means that our combined potential work power can be used more efficiently, because of easier planning and controlling each member's work. Yet this environment can only be constituted when every member acknowledges their responsibilities and plans their schedule according to it. We believe that all of the group members obtain that merit. Also three people can reach a consensus more easily, therefore disputes over the project’s direction and design potentially will be lower.

For the optimal on the go communication and collaboration environment concurrent technologies offer us a number of settings. Our WhatsApp group is the most useful of all, we can easily text our requests if there is, or notify the fellow group members in project related topics. We frequently met physically for easy communication, however that is not feasible at all times. Discord becomes involved in online communication. GitHub is useful for putting our coding efforts into a collective repository. Also GitHub enables us to view each group member's contribution, in that way we can give feedback more easily and thanks to the collective repository we can stipulate the project's progression more easily. For the gathered report writing, we use Google Docs, in this way we edit the same documentation, and put our commentaries. Other than technologies, we have done pair programming for efficient problem solving in the coding part.

Technological environments in the project:

- WhatsApp
- Discord
- GitHub
- Google Docs

6.3 Taking lead role and sharing leadership on the team

Picking a leader is essential, when solving a problem, because leaders are the ones with the plan. Without a plan, problem solving becomes chaotic, hard to solve. In *section 6.1,* our responsibilities for the projects have been established, in a way those responsibilities are problems to solve, thus needing a leader. Following are the leaders of each project related work.

Project Related Work	**Leader**
Initial planning, project specification	Enver Yiğitler
Analysis Report	Ant Duru
High-level design and frontend	Atay Kaylar
Machine learning models	Enver Yiğitler
Bug Fixing	Atay Kaylar
Low level design	Enver Yiğitler
Database systems	Ant Duru

Table 3. Leaders for each task

Following are the responsibilities of a leader:

- Identifying requirements
- Establishing goals and objectives
- Noticing the deadlines and informing the group members about them
- Distributing the work among group members
- Monitoring group members work

7. Glossary

Detailed dictionary of the report can be found in *section 1.3.*

8. References

[1]Institute, Mpower A.S.C.E.R.T. “Language Learning Leads to Improved Personality.” ASCERT, ASCERT, 29 Jan. 2020, https://www.ascertinstitute.org/post/language-learning-leads-to-improved-personality.

[2]Gobler, E., 2021. Best language learning app 2021: Top 7 apps compared | ZDNet. [online] ZDNet. Available at: <https://www.zdnet.com/article/best-language-learning-app/#:~:text=Duolingo%20has%20become%20the%20most,methods%20to%20help%20you%20learn.> [Accessed 20 December 2021].

[3]“Why I’m Quitting the Japanese Duolingo Course (an Honest Review).” Www.youtube.com, www.youtube.com/watch?v=jf-SbSfiXn4&t=480s&ab_channel=Livakivi. Accessed 20 Dec. 2021.

[4]Wikipedia contributors. "Kató Lomb." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 15 Nov. 2021. Web. 20 Dec. 2021.

[5]"Redis." *Vikipedi, Özgür Ansiklopedi*. 23 Tem 2021, 14.07 UTC. 21 Ara 2021, 16.40 <//tr.wikipedia.org/w/index.php?title=Redis&oldid=25855242>.

[6]"Swift (programlama dili)." *Vikipedi, Özgür Ansiklopedi*. 3 Eki 2021, 12.46 UTC. 21 Ara 2021, 16.42 <//tr.wikipedia.org/w/index.php?title=Swift_(programlama_dili)&oldid=26281826>.

[7]"JavaScript." *Vikipedi, Özgür Ansiklopedi*. 17 Haz 2021, 09.32 UTC. 21 Ara 2021, 16.43 <//tr.wikipedia.org/w/index.php?title=JavaScript&oldid=25644548>.

[8]By: IBM Cloud Education. “What Is Machine Learning?” IBM, 15 July 2020, https://www.ibm.com/cloud/learn/machine-learning.

[9]"SQLite." *Vikipedi, Özgür Ansiklopedi*. 8 Kas 2020, 13.05 UTC. 21 Ara 2021, 16.44 <//tr.wikipedia.org/w/index.php?title=SQLite&oldid=23981058>.

[10]"PostgreSQL." *Vikipedi, Özgür Ansiklopedi*. 9 Eki 2021, 23.54 UTC. 21 Ara 2021, 16.45 <//tr.wikipedia.org/w/index.php?title=PostgreSQL&oldid=26309590>.

[11]"JSON." *Vikipedi, Özgür Ansiklopedi*. 3 Eyl 2021, 13.01 UTC. 21 Ara 2021, 16.45 <//tr.wikipedia.org/w/index.php?title=JSON&oldid=26128184>.

[12]"EPUB." *Vikipedi, Özgür Ansiklopedi*. 10 Mar 2021, 17.39 UTC. 21 Ara 2021, 16.41 <//tr.wikipedia.org/w/index.php?title=EPUB&oldid=25087562>.

[13]"WebPub - Crunchbase Company Profile & Funding." *Crunchbase*, https://www.crunchbase.com/organization/webpub.

[14]Simard, M., Plamondon, P. Bilingual Sentence Alignment: Balancing Robustness and Accuracy. Machine Translation 13, 59–80 (1998). https://doi.org/10.1023/A:1008010319408

[15]Stanborough, Rebecca Joy. "Benefits of Reading Books: For Your Physical and Mental Health." Healthline, Healthline Media, 15 Oct. 2019, https://www.healthline.com/health/benefits-of-reading-books#alleviates-depression.

[16]"Keep Em Reading: The Importance of Audiobooks for Dyslexics." Keep Em Reading! The Importance of Audiobooks for Dyslexics | Dyslexia Help at the University of Michigan, http://dyslexiahelp.umich.edu/answers/ask-dr-pierson/keep-em-reading.

[17]ReadBeyond, Padova. What Is an Audio-Ebook?, 15 Apr. 2014, https://www.readbeyond.it/audioebooks.html.

[18]Longanecker, Chuck. "Why You Should Use a Subscription Business Model." Entrepreneur, Entrepreneur, 19 Mar. 2015, www.entrepreneur.com/article/243573.

[19]Rich, Emma. "The environmental impact of Amazon's Kindle" CleanTech Group, LLC, 2009, https://gato-docs.its.txstate.edu/jcr:4646e321-9a29-41e5-880d-4c5ffe69e03e/thoughts_ereaders.pd