Design Studio 3

Discussion 9am Team #5

01 - EXECUTIVE SUMMARY

Our team is presenting a software offering we call DesignTeam that will assist other software developers/designers to collaborate on the specifics of architectural design choices. There are a plethora of options in the design community to choose from. We are aiming to distinguish ourselves from other competing solutions by offering an online collaboration/development tool that allows designers to work together in selecting from the various architecture platforms and/or adding to the list of architectural design solutions based on the developers' needs. As a team begins to brainstorm on our forum and keywords are recognized by the application, suggestions for supporting architectures will be provided. Instead of making use of, or mimicking, popular diagramming tools and social networking environments, we intend to have a tool narrowly focused on the architectural design of software. We plan to have our users' focus continually directed toward architectural design decisions via the interface of our software solution. We want developers to have a go-to place (our design software) where they can easily access a forum/design environment where they can interact with other software developers in their respective teams and quickly come up with architectural design solutions. With DesignTeam, we call this interactive development environment the 'virtual board.' On this board we seek not only to offer recommendations for architecture design decisions, but have our system record new decisions made by those who participate in the design of software so that those decisions can be recommended to future architects as well.

02 - AUDIENCE AND OTHER STAKEHOLDERS

Below is the description of our potential users and stakeholders as they either will directly use our software with supporting tools and features on the application to develop architectural designs for their projects or learning purposes, or are involved in the process of software developing, designing and testing. Based on the role and involvement of each type of DesignTeam's users, their interactions with the software will be listed.

Concerning our potential users, we envision them as:

- ➢ **Software Designers/Developers/Seniors/Architects**
  - ○ Due to their need in presenting and designing an architecture of a software system or project. Software developers/designers will be one of the major users of DesignTeam since they would utilize our software's features and functionalities to create the architectural designs and diagrams. Another general usage of our software in the software developers' role is as a virtual environment to share thoughts and communicate ideas on the design decisions and get recommendations with appropriate architectural styles of their system.

- ➢ **Software Instructors/Experts**
  - ○ Due to their role in providing the instructions and demos on the software architectural designing process. Software instructors and experts would be people who directly utilize the student-mode feature of DesignTeam to demonstrate and teach general learning and student users of our software on different architectural styles and methods.

- ➢ **Students/Learners**
  - ○ Due to their need in learning about different styles of software architecture or joining to make architectural design decisions for their projects. Students and learners will utilize the student-mode feature of DesignTeam to learn about many different architectural styles and how to make an architectural diagram for a system through provided step-by-step instructions and demos from our software engineering and designing experts.

- ➢ **Project Managers**
  - ○ Due to their need in monitoring the designing and developing process of current projects on DesignTeam. Project managers will have access to the history of each stage of the architectural designs recorded on our software application.

- ➢ **Documenters**
  - ○ Due to their role in providing and documenting the requirements of current software projects. Their suggestions would be recognized and used to generate recommendations and supporting tools of appropriate architectures. Documenters will also utilize DesignTeam to notify developers/designers about changes in their software's requirements so they can appropriately modify or do maintenance of the architectural designs.

And concerning our potential stakeholders, we have ascertained that they will be comprised of the following:

- ➢ **Customers/Users**
  - ○ Due to their role in being the end users of the products built upon the architecture that developer and designer create on DesignTeam app. Customers/Users might also be involved in the software development process for building the product's requirements, testing and getting feedback. Their involvement might affect the design decisions for the development and design team.

- ➢ **System/Software Testers**
  - ○ Due to their need in obtaining and observing the design decisions and architecture of the product in order to drive test cases for the system's implementation that might manifest as errors and potentially become failures in the future.

- **System Engineers**
    - Due to their need in ensuring the software system works as a whole by analyzing the design trade-offs, systematic verification, validation, and requirements. They would need to analyze and observe the design and architecture of the software application to verify the overall management of the system by checking whether the subsystems, parts are working well together.

- **Software Maintenance Engineers**
    - Due to their need in gathering existing requirements, the building block and architecture of the current software project in order to analyze, make changes, updates and modifications to the software design to meet users' need and satisfaction.

03 - GOALS, CONSTRAINTS, ASSUMPTIONS FOR THE OVERALL DESIGN SOLUTION

- **Goals**
    - **DesignTeam should be developed to serve both beginners and professionals:** DesignTeam is developed to tackle some of the difficulties and hardness of software design for our users. DesignTeam aims to serve both professionals who already have deep knowledge and experience about designing an architecture for a system with our supporting tools and learners who need instructions and guidance on how to construct a software architecture. Regardless of the user's background, we aim to provide help and assistance to our users with well-developed features in their journey of making design decisions and building software architectures.

    - **DesignTeam will provide instructions of software design for beginners/learners:** Students and new learners who are not familiar with software design might face difficulties with starting a software project due to its complexity, especially when it comes to software architectures. DesignTeam aims to provide guidance and assistance for students/learners to learn basic software design through our feature: student-mode. Student-mode aims to assist learners and students on their architecture learning purpose. This feature will be used by students and learners who have little to no experience on software architecture. They can utilize this mode to learn and play around with different architectural styles.

    - **Allows multiple designers/developers to work together on a software design project:** As software architecture design is a hard and long process. DesignTeam allows multi-participations through live videos in the software design process in order to provide accommodation to different representations and design artifacts for developers and designers. This will help developers and designers work together through a shared environment where they can communicate, think about the problems, generate ideas and verify with other people about issues and thoughts of the current project.

- **DesignTeam will be able to generate recommendations for architecture design decisions:** Recommendations of appropriated architectures would provide users references, directions and help them think about some possible approaches to build their software applications.

- **DesignTeam offers both personal and shareable approaches to software architecture designs:** One of DesignTeam's goals is to provide our users with freedom and choices in building their software application projects either personally or shareably with others. Users of DesignTeam will be able to develop a basic software design with/without interactive instructions on their own or they can also be able to collaborate with others in building and developing their projects. We respect each user's choice and aim to serve them with best preferences and approaches.

- **Constraints**
  - **User must provide a login to access DesignTeam:** To access DesignTeam, user must create an account and provide a login. As the system will keep track of the user's software design history and current projects, as well as their private information and data, a login is required for accessing DesignTeam. Users would be able to view their current works and projects (with records and history of different stages) on their account.

  - **DesignTeam must provide step-by-step and visual instructions to assist student mode:** A student-mode should be developed to serve learners and students who have little to no experience and familiarity with software design. It should provide students with interactive and step-by-step instructions on how to build a basic software project. DesignTeam must offer students control and freedom in pausing, accelerating, going back and forth at different steps, directly working with and building a diagram during the instructions.

  - **DesignTeam supports different types of architectural designs and styles:** DesignTeam should provide a variety of different architectural design styles for users to choose from (either for making or learning). Supporting tools to do brainstorming, draw diagrams and building architectures should be developed to serve different architectural design styles.

  - **DesignTeam must have a system history and record for current design projects, future recommendations and users' reviews:** The system must keep track of records and versions of the design architecture made by developers and designers at different stages of development so the architect and developers can look back on their existing works. Developers will be able to continue building and changing their work at any of the recorded stages. These records would also be used to generate future recommendations for the architectural designs.

- ○ **DesignTeam must be accessible via any public network connection:** In order to provide portability, availability and accessibility for users to their software design projects on the application, DesignTeam must be accessible via any public network connection anywhere at any time.

- ○ **To access DesignTeam, a user's device must be connected to a stable Internet or WiFi connection:** In order to ensure a smooth process of software designing and keep up with the progress, users are required to have a stable connection to the Internet or WiFi. During the learning process in student-mode, interruption or disconnection could cause loss of data and records from the process.

- ○ **DesignTeam must provide a virtual forum/environment for multiple users working together:** A virtual board is required to provide a forum/environment for users to develop a software project. Multiple developers and designers should be able to join, work together and use supporting tools on the virtual board.

- ○ **DesignTeam must be developed to operate on various platforms including web-based, mobile, and PC:** In order to provide users with accommodations and accessibility to their software projects, DesignTeam must be able to operate on various platforms. Users of DesignTeam should be able to access the software through any popular web browser, mobile platform and PC.

➢ **Assumptions**

- ○ **All users provide a stable connection to DesignTeam to ensure a smooth architecture design process:** A stable connection to DesignTeam should be ensured to have full access and update from the progress of software designing.

- ○ **Recommendations for architectural styles and designs are only for references:** Recommendations from our software detecting tool for architectural designs might not be perfectly applicable to the current software projects of users. These recommendations are used for providing guidance and generating referenced approaches to the architecture of user's projects.

- ○ **One member of the project team could modify and make changes to the architectural designs in the interest of other members:** Since every member of the development/design team has access to different stages and records of their projects, each of them has the control to change or modify the system's current design version. Users should acknowledge that changes made by one member on the current version will be applied to other members as well.

- **All users have sufficient technical knowledge and experience:** Usage of DesignTeam would require users to have sufficient technical skills since they would need to know how to use supporting tools in designing a software application.

- **Beginners/learners are engaging in the student mode of learning:** Software designing is hard and it contains large and heavy weighted knowledge. Hence, the learning process of software design and architecture would require lots of patience and time. Beginners/learners will be guided through detailed and step-by-step instructions on student-mode from DesignTeam and their participation will determine the success of learning.

- **Users should acknowledge that there might be minor differences in the design notations developed on DesignTeam with others across the Internet:** Design notations for different diagrams from the supporting tools might have minor differences from the ones on the Internet. However, instructions and explanations on these design notations will be provided in help feature and student-mode.

04 - MAIN DESIGN

- **Application Design**

  Brainstorm/virtual board

  - Our application shall provide a virtual board brainstorming tool for users to start off their application project. Users with their team members shall be able to begin either with an empty canvas or choose one of the brainstorming tools such as “mind-maps”, “morphological chart”, etc, provided by our application to brainstorm, analyze, organize and connect their thought processes. Teams will be able to work collaboratively on the virtual board to capture their main ideas and enter crucial keywords for our application to identify and later be used to generate architecture design tool recommendations.

  Generate recommendations for architecture design decisions

  - Our application shall provide and generate architectural design tool recommendations based on keywords gathered from users brainstorming map. Once users are satisfied with their brainstorming process they can click NEXT and the system will capture the current brainstorm page and categorize the type of application to generate appropriate architecture design models.

  Collaborate on the specifics of architectural design choices

  - Our application shall provide users with a wide variety of online interface/collaboration/development tools that allows designers to work together in selecting from the various architecture platforms and/or adding to the list of architectural design solutions based on the developers’ needs. Our application will offer real time collaboration where each member of the team shall be able to view each other's cursor and see new edits simultaneously.

Live conversations and Video conferences

- Our application shall provide users with live conversation functionality where team members could join-in together through chat or voice/video call while working on their products software architecture design. This will allow smooth communication between team members and reduce communication difficulties making collaboration more efficient and effective. This feature shall also support users to work on their architecture while video conferencing at the same time.

Notification of editing

- Our application shall provide users with version history of their architecture model documents. Users will be able to view past versions and edits made by which user and also restore the certain versions. The application shall allow users to leave direct comments by highlighting areas of their diagram for other users given access to view. The project owner shall be able to choose to be notified about edits made on diagram and when other members are currently online.

Learner mode

- Our application shall provide a step-by-step learning mode for users that are not familiar with software architecture. While in learner mode, the system shall support user learning step-by-step through videos, tutorials, and examples from senior developers. Each recommended diagram will contain detailed explanations of the benefits and targets of the model. This feature will give the students a broad understanding of the field and explain how each diagram works in real-world applications.

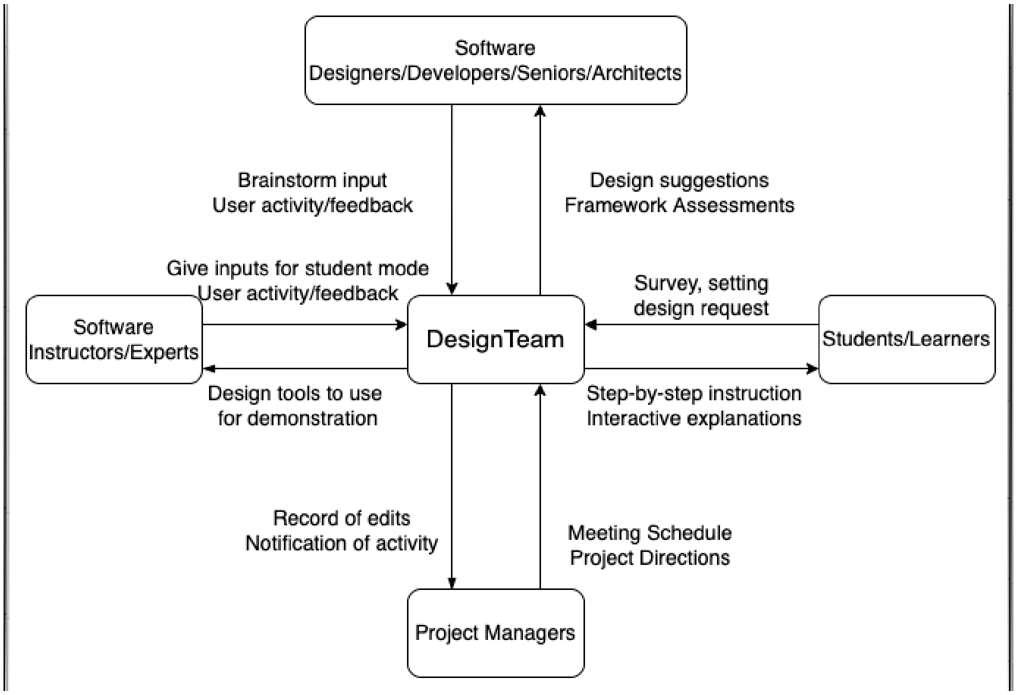
Interactive Screencast

- The application shall provide interactive screencast features for students in learner mode. The system shall make use of call-to-action buttons and let users know what they should do next. The interactive screencast feature shall also be equipped with voice over explanation to better assist user interaction. The interactive screencast function shall also provide users with customizable choices such as voice volume, mute, or speed-up/slow-down.

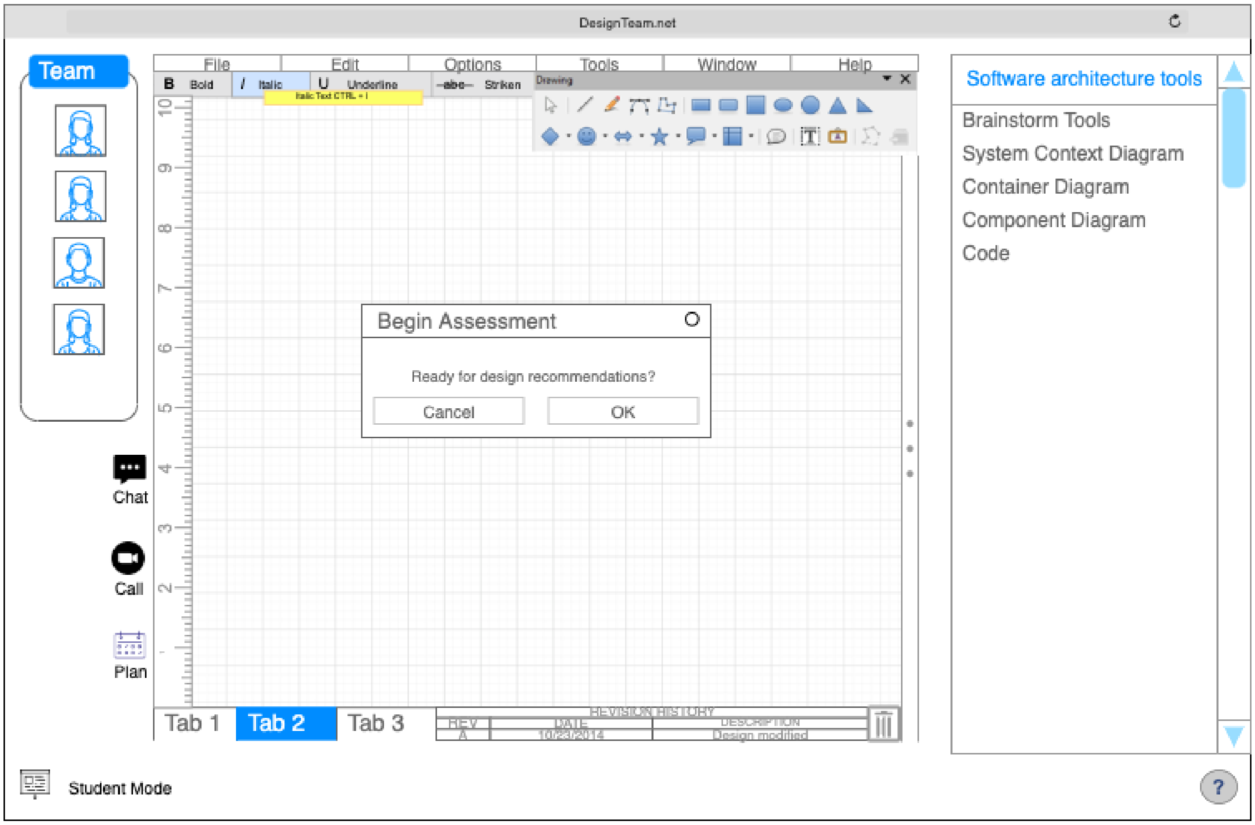
Make improvements based on user activity/feedback

- Our application shall ask users to grant our system permission in tracking their usage decisions in order to make improvements. If granted permission, our system shall record new decisions made by those who participate in the design of software so that those decisions can be recommended to future architects as well. Our application shall also provide users with feedback surveys to address any problems for future improvement considerations.

Context Diagram:



➢ Interaction Design



Begin Assessment

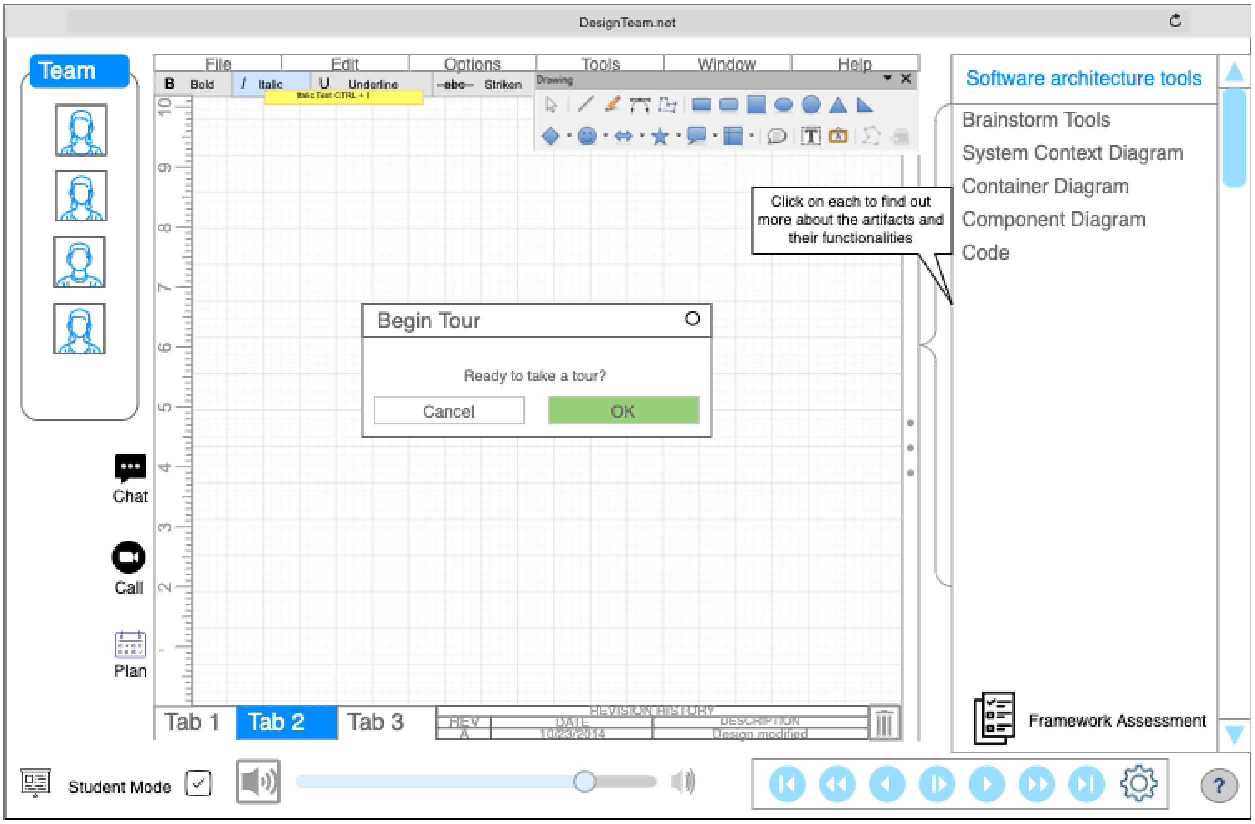
Ready for design recommendations?

Cancel OK

Figure 1: Virtual Board in Professional Mode

Main interaction:

- Upon login to the web-page users will be prompted to create a new project. The project directory created will be able to be accessed by multiple users if given the sharing link. The users of the project will appear in the top-left section of figure 1 whenever they are active on the page. Users will be presented with a blank virtual board located in the center of the page. When different users are working simultaneously, each user's cursor will appear on the canvas with their profile names. The tool bar presented on the top of the canvas can be utilized by users to begin their initial brainstorming design and navigate DesignTeam. After the brainstorming process, users can request a brainstorm design assessment and DesignTeam will capture the keywords and components of the project and present users with design recommendations. The menu on the right of the virtual board is a simplified example of the software architecture tools provided to users. Users will be able to click on any category and select tools they would like to add to their design.
- Software Architects can utilize the group functions located on the bottom left hand corner of DesignTeam for communication purposes. The chat box is where all members with access to the project can communicate and send text messages, voice messages as well as files. The Call box is where all members with access can join-in voice calls or video calls. The chat and call box can be minimized to enable multitasking where users can have a call in the background while working on the project. The plan box can be used to mark date and time for meetings and to plan the outline of the project.
- The version history box located at the bottom right corner of the virtual board can show new changes, most recent editions and past versions of the document. Users can click into the version history and view past versions or restore versions.



Click on each to find out more about the artifacts and their functionalities

Ready to take a tour?

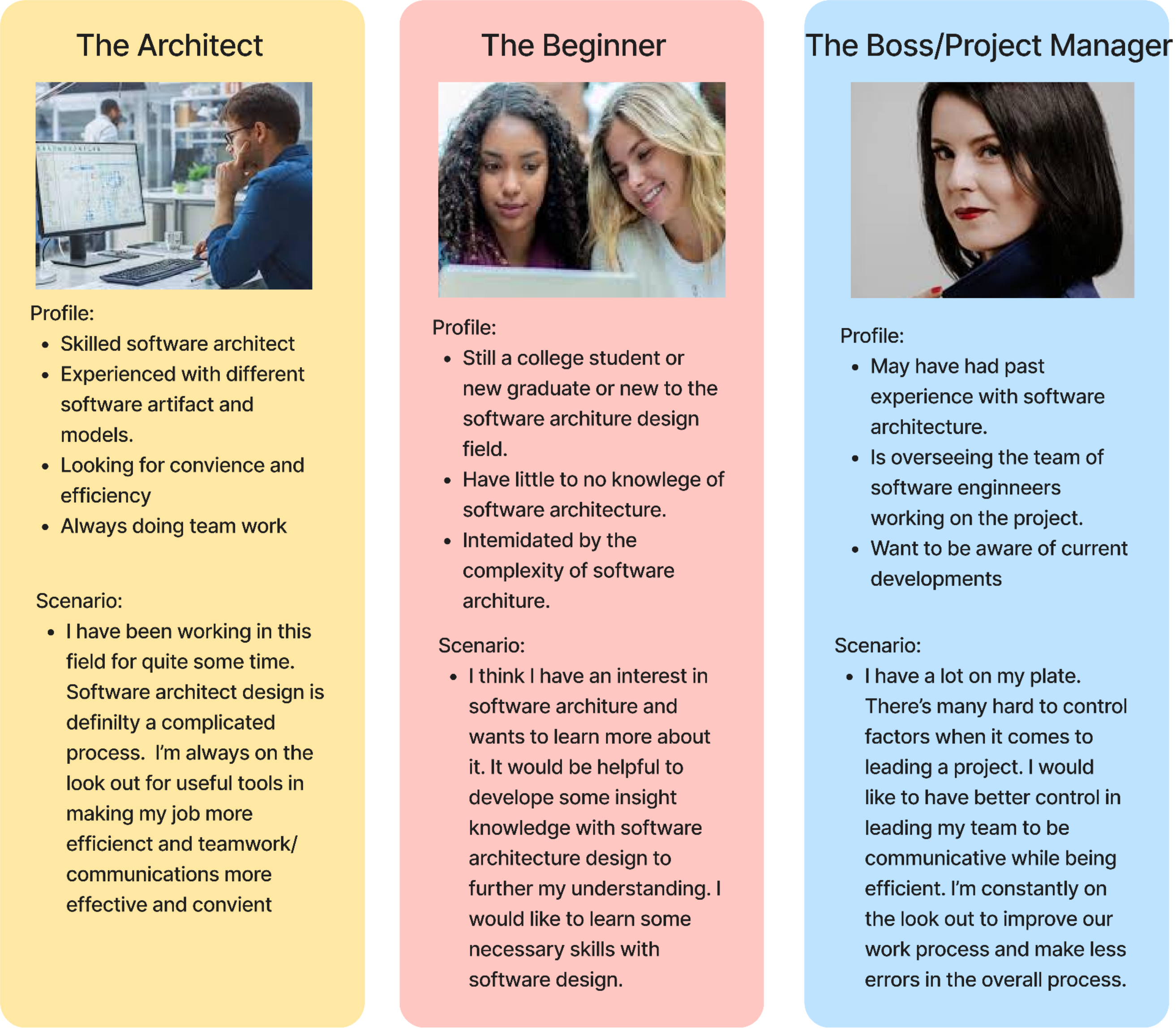
Figure 2: Virtual Board in Student Mode

Student mode additional interaction:

- Student mode supports all above interactive features of the professional mode with the addition of interactive screencast and voice over feature to improve user usability.
- Upon logging in to their account. They can begin new projects. When student/learner mode is activated learners will be prompted to first take a tour of DesignTeam to learn more about the feature and functionality. The basics of software architecture will be introduced here. After the basic tour users can continue to move onto the next step which is the start of the framework assessment located in the bottom right corner. The framework assessment will ask users to input keywords of the proposed project and other information to better understand the direction of the project. Then the brainstorm process will be explained and introduced and users can either start mind mapping on the virtual board or use one of the provided templates. The next steps will be the save as professional mode with the addition of voice explanations of the different recommendations listing out the strengths and weaknesses of the model.
- Some additional features include adjusting the learner mode voice reader. Users may choose to turn off the voice explanation and only rely on text bubbles and other interactive features in text. Users can also choose to speed up or slow down the voice guide or replay.

- Users can utilize the question mark located in the bottom right corner of the page to look up relevant topics and helpful directions.

Personas:



The Architect

Profile:

- Skilled software architect
- Experienced with different software artifact and models.
- Looking for convience and efficiency
- Always doing team work

Scenario:

- I have been working in this field for quite some time. Software architect design is definilty a complicated process. I'm always on the look out for useful tools in making my job more efficienct and teamwork/ communications more effective and convient

The Beginner

Profile:

- Still a college student or new graduate or new to the software architure design field.
- Have little to no knowlege of software architecture.
- Intemidated by the complexity of software architure.

Scenario:

- I think I have an interest in software architure and wants to learn more about it. It would be helpful to develope some insight knowledge with software architecture design to further my understanding. I would like to learn some necessary skills with software design.

The Boss/Project Manager

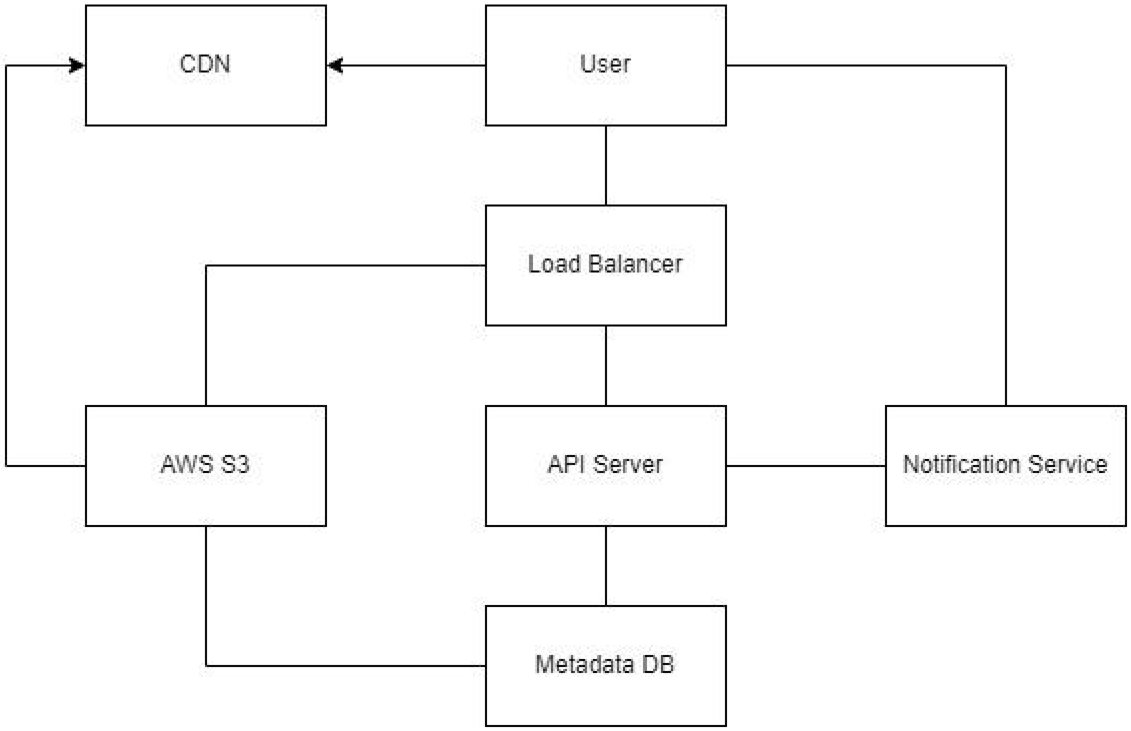
Profile:

- May have had past experience with software architecture.
- Is overseeing the team of software enginneers working on the project.
- Want to be aware of current developments

Scenario:

- I have a lot on my plate. There's many hard to control factors when it comes to leading a project. I would like to have better control in leading my team to be communicative while being efficient. I'm constantly on the look out to improve our work process and make less errors in the overall process.

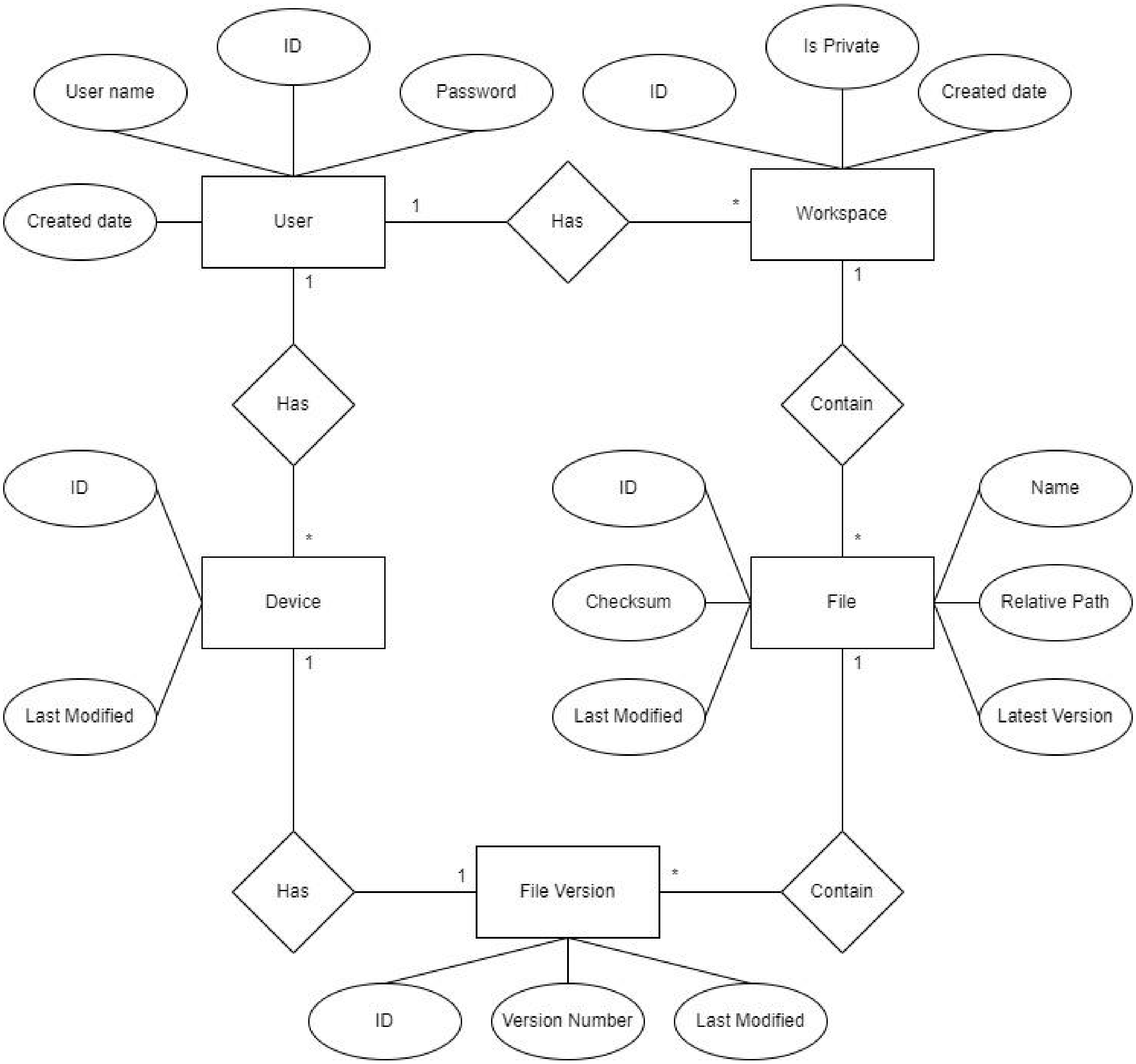
➢ Architecture Design



The figure above depicts the interdependence of our system's components. Bidirectional directions are represented by lines, whereas single directions are represented by an arrow on a line. Our system includes:

- ○ **Load balancer:** It is a critical necessity since, with the increase of online traffic, a single server setup may bring the entire system down. A load balancer can assist in distributing incoming traffic evenly across servers designated in a load-balanced set. The private IPs are also hidden in this configuration, and users cannot directly reach the servers, which contributes to improved security. If one of the servers goes down or takes too long to respond, a load balancer may divert the incoming requests to other servers, avoiding bottlenecks and enhancing service availability.
- ○ **API Server:** It handles requests from the front-end side, including authentication, data retrieval, modifying data… Spring MVC with Boot is our first choice for developing the DesignTeam project and hosting it on our AWS server since it is a lightweight framework that enables auto-configuration and supports the JDBC framework, which improves productivity, reduces mistakes, and eliminates SQL injections. Following the MVC pattern also helps us to make complicated application development much more manageable by dividing it into three components:
    - ■ **Models**: the code often mirrors real-world objects, and data models, and is used to keep raw data from the database before returning it to the client. For the DesignTeam project, we'll have five models in total:

    - **User**: holds all of the user's fundamental information, such as user_id, user_name, password, and created_at.
    - **Workplace**: this class will act as a folder to store all diagrams and documentation of the projects. To do so, this class must consist of information about workplace_id, owner_id, is_private, and the created date.
    - **File**: this class contains information regarding files, including file_name, relative_path, latest_version, checksum, workplace_id, created_at, and last_modified.
    - **File_Version:** this class stores the version information of a file, including the version number, last modified date, device id, and file id.
    - **Device**: This class consists of the devices' information, including device_id, user_id, and last_logged_in_at, which are used to check the authorization preventing hackers from exploiting data.
- ■ **View:** The user interface is used to display the current state of models and re-render when the model state changes. Also, views are built using data gathered by the model component, however these data are acquired indirectly through the controller, therefore the view only communicates to the controller.
- ■ **Controllers:** The controller class is responsible for handling RESTFUL API requests, preparing obtained data from the database, and providing the result back to the client as JSON. In order to hide the complexity of the CRUD operations in the underlying storage mechanism, we will use the DAO pattern, the Data Access Object pattern, which allows us to isolate the application layer from the persistence data tier and decouple data models from the database.
    - For each model in the MVC pattern, we will construct an interface that has method signatures and all queries needed to access data, as well as an implementation class that does the logic to extract/update data from the database.
    - In the controller classes, we will inject the appropriate DAO that we’ve created in the last step and implement methods performing all of the access to the database.



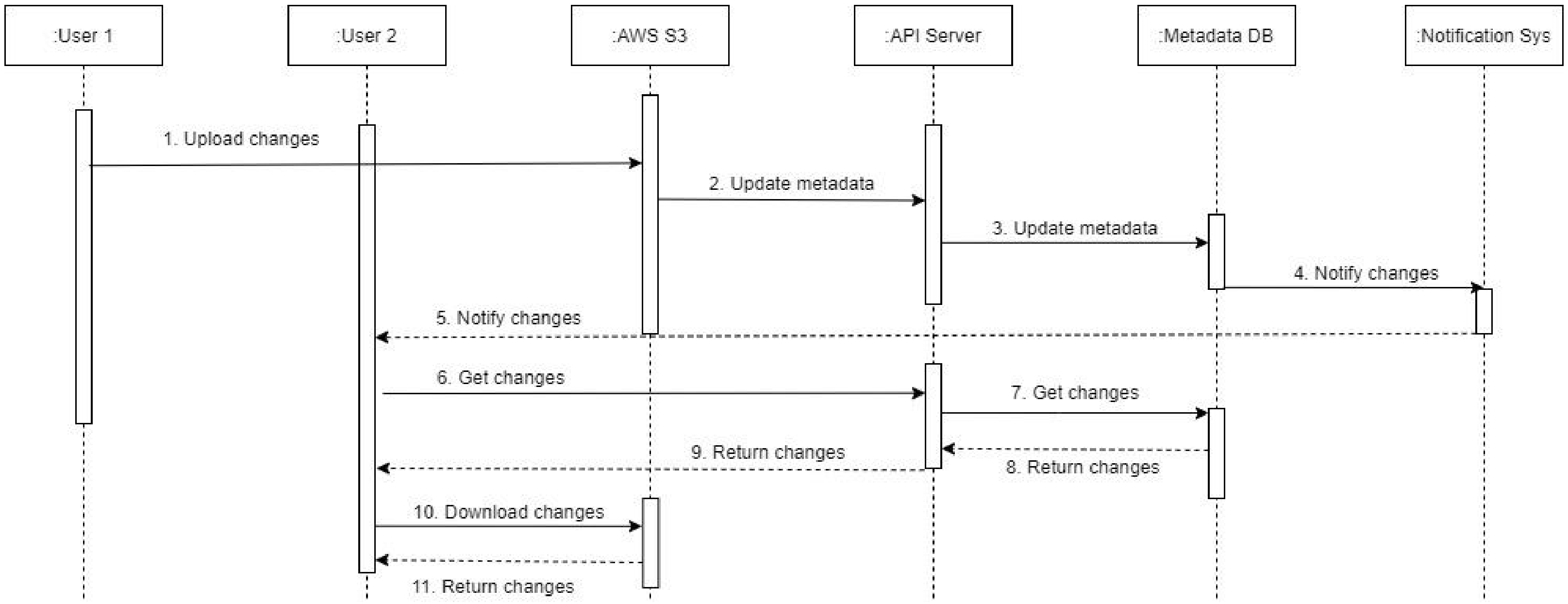
Entity Relationship Diagram for the DesignTeam project

- **Metadata DB**: It saves the models in the ER diagram, including User, Workspace, Device, File, and File Version. We will use Postgresql as a database management system because it aids in the preservation of data integrity, and data correctness reduces data redundancy to a minimum or zero, data scalability, data flexibility, and simplifies the use of security mechanisms. Also, we want to achieve strong consistency — ACID properties, which stand for Atomicity, Consistency, Isolation, and Durability so that the virtual board can be seen synchronously between users.
  - Database replication will be used in this project because it can improve reliability and availability with a master-slave model. In most cases, a master database is only available for write operations, while a slave database receives copies of the master database's data and is available for read operations. For all operations, such as insert, remove, and update, the

master must receive all data modified to replicate continuously and send it to the slaves. Supporting database replication can improve performance significantly by allowing multiple queries to be executed concurrently. Replicating data, furthermore, can enhance availability and reliability because when one of the database servers crashes or is destroyed by natural disasters, all data will be preserved and accessible from everywhere.

  - **Notification Service**: It is a publisher-subscribe service that allows users to receive notifications and register for notifications. Basically, it will notify users when there is a change made by another user. It, additionally, allows users to see changes over time.
  - **AWS S3**: Because AWS S3 supports same-region and cross-region replication, which increases data availability and durability, here is where virtual board files, instruction videos, and screencasts are kept. It ensures that data will not be lost due to server failures or natural disasters.
  - **CDN**: this is where videos are directly streamed. Videos will be sent to CDN following the transcoding process. When a user requests a video, it will be delivered via the nearest server, allowing for low latency and a smooth experience.

- Virtual board:
  - A virtual board is a feature where users may work with others to produce architectural ideas that meet their specific requirements. Because the software must support a wide range of activities, such as drag & drop, select, copy, and paste components/objects, QT is the ideal choice for the users to have a seamless experience across platforms. Furthermore, Qt is a cross-platform application development toolkit that allows developers to produce excellent applications with uniform user interfaces for desktops, embedded devices, and mobile platforms. It provides developers with an excellent collection of tools for designing and producing exceptional apps without having to worry about platform requirements.
  - Because of the huge demand for working on web browsers, we will develop another version of DesignTeam using React so that developers may work anywhere they choose. It is, however, a stripped-down version of DesignTeam because the web app requires an internet connection to access, load, and back up data. Hence, this version will give the users almost the same experience, except for the offline mode.
  - Users may import and export the virtual board, as well as add it to the documentation, in addition to collaborating with other developers. If the big-fat and complex diagram was the scariest thing for developers when leaving the reference image in the docs, it's now simplified and easy to do with the virtual board; it allows users to create a minimal version of the diagrams, and when the user clicks on the mini diagram, it will zoom in and allow them to drag and see the details of it.
  - Sequence diagram of the data retrieval:



The diagram above depicts the interactions that occur when two users sketch diagrams on Virtual Board at the same time.

        - When the changes have been made, the client-side of user 1 will send an update to AWS S3.
        - Once the file is uploaded, the storage will update the metadata on the database via the API server.
        - The notification system will notify user 2 that there is a new update available to download.
        - Then, the client-side of user 2 will send a request to the API server to get the metadata of that file and download the changes from AWS S3.

- Interactive screencast:
    - Working along with the Virtual Board, the interactive screencast is a bonus for students/developers who are interested in learning more about architectural designs.
    - Using voice and scripts to make "real" animations so that learners may change the diagram before/during/after the instructions are necessary to provide learners with hands-on experience in developing each sort of diagram. Lua is a good option because it is much easier to use than C++ and is at a much higher level. The Lua engine has a small memory footprint and can be built-in runtime, allowing developers to edit/modify scripts fast and effectively without restarting the app or recompiling parts of the script. Lua is also very versatile since it can run on several platforms, including web applications. As a result, developing live animations will be easier and more approachable for both developers and users.
- Live chat and Voice calls:
    - These features are empowered by end-to-end encryption so that each frame of video calls or messages will be encrypted before sending to other ends. Public-key encryption is a great choice because it is asymmetric encryption,

which requires a pair of keys to encrypt and decrypt messages. This method can prevent hackers from guessing the key's length, and contents, especially the algorithms used to encrypt and decrypt messages and video calls. The basic steps in sending and receiving messages among users:

        - ■ Each user will create a pair of keys — a public key and a private key to encrypt and decrypt messages respectively.
        - ■ When the connection is established among users, each user will send a public key over a channel to all other users in a group chat.
        - ■ Right now, when users want to message each other or all people in a group, they will send an encrypted message using the public keys that they've got to encrypt the message before sending.
        - ■ When messages are delivered to other ends, the users will use their private key to decrypt the message.
    - ○ For video calls, we will apply the same strategy — using Public-key encryption to protect media from being leaked with a twist:
        - ■ When a user joins or leaves a video call, each user will create a new pair of keys and distribute them to all other users because we don't want the user who just left the call to receive any encrypted video frames and know what's happened after they left. With this solution, hackers or bad guys cannot exploit the media.

05 - ALTERNATIVES CONSIDERED

- ➢ **Alternative #1: A selection of pre-set architecture design layouts to choose from, created by the system's developers or third-party developers/individual developers.**

    - ○ Description: The *Virtual Board* will have a drop-down searchable column that contains the feature to search for pre-set architecture design layouts which will auto-populate once clicked on chosen. These pre-set architecture design layouts contain the most basic framework of an architecture design, and can be filled in with customizable elements using drag-and-drop method.

    - ○ Comparison of Approaches: There's no such similar features on the system just yet. This pre-set architecture design layout feature will help minimize the time to set up the framework of a new architecture design project. In addition, it gives the users an idea of what to expect on a specific architecture design they're looking to produce, and through following the pre-set architecture design layouts, they will get an idea of what to be focusing on and including on their pre-selected layout(s). Although this is a good idea, it may discourage users of the *Virtual Board* to be less creative on their architecture designs and more reliant on these pre-set layouts to begin their project designs. Furthermore, people tend to use similar pre-set layouts for certain types of design projects and this creates a duplicative environment where every architecture design on the software board will be very similar to each other in the framework aspect. Therefore, as we prioritize creativity over convenience, we decided to opt out of this feature for future updates.

- **Alternative #2: Allowing users/groups to upload their architecture designs to get ratings and feedback from other expert users on the virtual board (outside teams).**

  - Description: The *Virtual Board* will have a feature for users from different teams to upload their architectural designs, at will, onto the platform for other expert users (not beginner accounts or recently created accounts without any contributions) to give ratings and provide feedback to that design. The ratings will be out of 5 stars; 1 star being the worst rating and 5 star being the best rating, and feedback will be optional but highly suggested. This will help the creator of the architecture design to receive feedback for improvements on future updates to their designs. In addition, different teams can use these sample designs to begin brainstorming ideas for their own architecture designs.

  - Comparison of Approaches: This feature will be similar to the survey feedback feature for members within a team, but the main difference is that the feedback is open to all users on the *Virtual Board* (members from outside teams) and not only limited to the people within the team. Teams can get accurate and more diverse feedback from different teams, which will ultimately help them to make improvements and updates to their designs to best fit the general diverse audience. Since people tend to have biases toward certain things in life, users on the *Virtual Boards* will also have biases on certain types of design choices. Therefore, if a team doesn't like a specific architecture design, they will likely give a bad or negative rating to that design choice. In addition, different teams have different specialties in their own architecture design choices, and they may not be too familiar with another architecture design choice in order to give a comprehensive and constructive feedback/ratings. So, ratings from all users on the app may not be as accurate as we thought, and can't really help the designers to pull any meaningful insights rather than opinions and personal references. Therefore, we decided to opt out from the public rating and feedback system, and stick with the survey feedback within the team system.

- **Alternative #3: A team-lead mode / account with the privileges to assign sections of the architecture design to a specific team member to work on.**

  - Description: The *Virtual Board* system will permit a creation of a team-lead account within each team where the team-lead will have the privileges of a team leader: assign their team members to a specific task/section of the architecture design to work on, and provide additional instructions to the member. This will help the team members to stay focused on a specific task assigned to them, and makes sure everyone on the team works synchronously with one another to complete the design.

- Comparison of Approaches: This team-lead account feature is similar to the project manager role in a way that both positions involve managing of some kind. The difference is that team-lead roles manage people and tasks within their team, whereas project managers only manage the process of the project as a whole across multiple teams. The team-lead will have the ultimate power to design a team member's role and appropriate tasks that best fit their ability according to the team lead. Although this would create an organized task distribution system managed by the team lead, it prohibits the freedom to choose and pick tasks among other team members. The freedom to actively collaborate with one another and figuring out the strengths and weaknesses in each team member can't be performed under a singular team-lead control setting. Therefore, as we prioritize flexibility and the freedom to choose, we omitted the feature of a team-lead mode.

➢ **Alternative #4: Implementation of third-party design tools onto the Virtual Board for a smooth and seamless experience.**

- Description: The *Virtual Board* will allow third-party design tools to be imported on the system. Design tools like Figma, Adobe XD, and Sketch can be imported using log in to those third-party accounts. Once logged in, all design layouts will be available to the *Virtual Board* user to choose from and import into *Virtual Board*. Additionally, elements from third-party design tools can also be imported into *Virtual Board* for usage.

- Comparison of Approaches: The *Virtual Board* system does not already have this feature implemented. It can be convenient for the *Virtual Board* users to be able to import their already existing designs from the third-party design systems like Figma, Adobe XD, or Sketch, and continually work on those designs with other team members on the *Virtual Board* system. However, it would require time to acquire permissions and authorizations from the third-party vendors in order to implement this feature onto the *Virtual Board* system. Third-party systems' requirements and specifications can be conflicting with the already existing features and architecture of the *Virtual Board,* and that will not always be complementary to the *Virtual Board* system. Therefore, we decided to avoid adding third-party components and imports onto our *Virtual Board* system.

06 - ETHICAL ANALYSIS

During the development of this architectural design software, we took into account how the design may or may not have an ethical component to it. This particular design does not seem to have many moral implications. But if we understand that this is a collaboration platform where users interact with one another, we can find a few places where our software solution can help moderate discussions. However, with any social engagement platform, there will always be an opportunity for those who choose to be offensive to do so.

Another point of consideration is the language offerings of our product. Currently it is English only. However, if the software were to grow in popularity, we would seek to create a built-in language option selector.

As far discriminating against any particular group, we do not see this software doing so. It appears to be group-neutral with the exception of the language portion mentioned above. With our ‘student mode,’ we also made sure to make it easier for those who would make use of our software when they have limited knowledge of the subject matter.

The environmental impact of our design solution is negligible considering we would only be making use of a handful of servers to begin with and scale up if necessary.

Concerning those with hearing and visual impairments, we have yet to accommodate them via accessibility tools. Our hope presently is that already available third-party tools will indirectly assist those who need them, such as screen readers and voice-to-text options. We will include the ability to zoom in on diagrams and enlarge text as needed.

Our software will be provided for free for the student mode only version, but for the professional/enterprise capabilities we will charge a reasonable fee for our collaboration service. This is mentioned under the ethical analysis to show that we intend to be reasonable with the fees our service will charge.

We intend not to use any harsh or offensive language in all of our help menus, tutorials, dialogue boxes and so on. We desire a professional presentation of our language throughout the user experience of our software.

We want to narrow our decisions for cloud service to companies that are responsible environmentally. If our current choice, AWS for now, turns out to not be the best one, we will look elsewhere for a hosting solution.

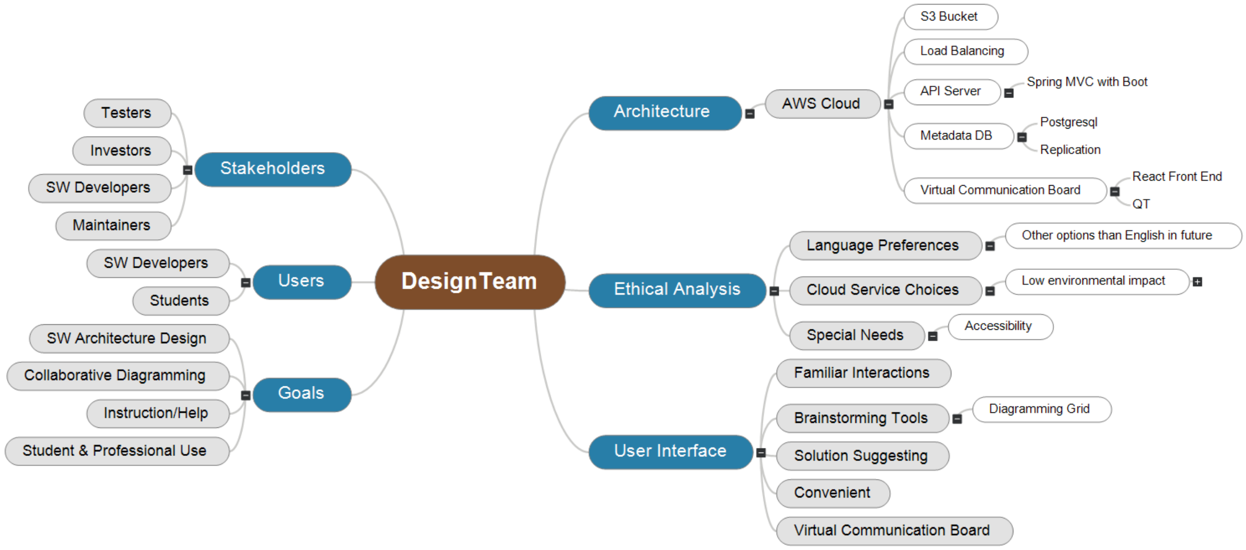
Future iterations of our design system will be examined for any potential ethical aspects that could affect users, stakeholders, and others.

07 - CONNECTION TO COURSE CONCEPTS

Our team employed many of the concepts learned in this course in our development of Design Studio 1, 2, and now 3. We have come to understand that design is a very repetitive and complex process. It requires multiple and ongoing iterations to bring a design to market. As a team we carried on with a problem/solution dialogue that involved many, but not all, of the design lessons we learned in class. For DS3, we found that the emphasis on architecture from DS2-2 helped out significantly. We have kept our focus on potential users and stakeholders in our design process. Our team has also become acquainted with the application/interaction/architecture pattern of development. Goals, constraints, assumptions, decisions, opinions, ideas were repeatedly emphasized throughout the quarter, and this did have an impact on each one of us as we discussed how to move forward. The analyze, synthesize, evaluate circle of design was not mentioned explicitly by our team, but it appears to have been something we adopted during this course. Concerning one the first things we learned about expert practices is that expert designers prefer to work in teams rather than alone. As a team developing the design for our product, we have noticed that we are much more productive when we are in regular communication with each other. We would also branch out with multiple concurrent dialogues between team members focusing on a particular aspect of the design.

08 - MIND MAPPING

The mind map below was the result of receiving input from all 5 of our team members in the design of DesignTeam, our architecture design software solution. Each one of us provided helpful insights, and each member had areas of experience and knowledge that contributed to this overall design. The mind-mapping process helped us to expand our ideation into other previously unknown areas of thought. This diagram also generally provides an overview of what is described in the above sections of our design plan.


DesignTeam
Stakeholders
Testers
Investors
SW Developers
Maintainers
Users
SW Developers
Students
Goals
SW Architecture Design
Collaborative Diagramming
Instruction/Help
Student & Professional Use
Architecture
AWS Cloud
S3 Bucket
Load Balancing
API Server
Spring MVC with Boot
Metadata DB
Postgresql
Replication
Virtual Communication Board
React Front End
QT
Ethical Analysis
Language Preferences
Other options than English in future
Cloud Service Choices
Low environmental impact
Special Needs
Accessibility
User Interface
Familiar Interactions
Brainstorming Tools
Diagramming Grid
Solution Suggesting
Convenient
Virtual Communication Board