Vision Document: Provides an overview of the project's objectives, scope, and stakeholders' needs.

**1 – Background**

* Communication is a vital skill in any collaborative discipline, and software engineering is no different
* Effective communication can make or break a project and prevent Wall Street headlines
* You hear all the time that communication is everything, but what does effective communication look like in a corporate environment?
* Want to make a personal observation: Text based communication mediums are less effective than video/calls/in-person conversations
  + Less disruptive train of thought/context switching (hard to have two conversations at once, more focus)
  + Faster conveying of information
  + Can read nuanced body language/tone to help drive the conversation (if the other person has concerns, can see/hear it if they don’t say it directly)
* In a corporate environment, the most common personifications of this are simply walking up to someone’s desk and having a conversation, or going into a meeting room
* With open office conversations, you may be talking about sensitive information that other coworkers aren’t supposed to hear, or are maybe doing a code review where you may have to give constructive criticism, and don’t want the receiver to feel any embarrassment with other people hearing it, kind of like your mom reprimanding you in front of your friends
* So that leaves the second option of conference rooms as the preferred method, but unfortunately at Buzzword Software Solutions we currently don’t have a system to properly facilitate this, as you may walk into a conference room that’s already in use, have trouble finding a room that can accommodate all participants of the meeting, or take up a room that was planned to be used/needs to be used now for more urgent matters.
* Which is exactly what our project is at Team Green, a system to manage the use of meeting rooms so teams can find and claim rooms to have a proper space to communicate and iron out the requirements/directions of their projects

**2 – Business Opportunity**

* Productivity increases as stated with teams having the appropriate environment to discuss project goals, or maybe talk about sensitive information
* Flexibility increases with in-office workers able to better collaborate with hybrid-remote workers
* Profit increases with better communication leading to quicker deadlines being met and an overall increase of the efficiency of the company

**3 – Business Goals and Objectives**

* We simply want an organized system that fosters better communication between employees (*this statement can easily be changed its trash*)
* We will build an extensible system so that more features can be added into future version without significant changes occurring throughout the project

**4 – Success Criteria**

* Increased in engagement amongst employees
* Increase in communication effectiveness
* Increased productivity/faster project completion, as fewer follow-up questions will need to be asked
* Fewer conflicts with booking rooms or scheduling meetings

**5 – Customer/User needs**

* Some way to schedule meetings in a way that’s easy for employees and can be overseen by managers
* Without a proper way to reserve rooms, meetings can be missed or dropped, rescheduling meetings can cost companies time and money.

**6 – Business Risks**

* Minimal infrastructure cost, database setup and hosting
* Ongoing support and a maintenance team
* Time and resources to train employees on the new system

**7 – Vision Statement**

* We are essentially proposing a scalable communication infrastructure for our coworkers that would improve productivity and enhance flexibility, which will over time increase profits (*can also be changed its trash)*
  + [7 Ways Video Conferencing Can Elevate Collaboration | iVideo](https://www.ivideo.com/7-ways-video-conferencing-can-elevate-collaboration/)
* As more acquisitions are made, the system can be integrated into new offices

**8 – Major features**

* Employees of various seniority are the users
* They must be able to:
  + Add/update room reservations
  + List open room reservations
  + Search for rooms by seating capacity
  + View who reserved a room for what timeframe
  + Schedule video conferences that also connect to other video conference rooms across the globe
* Executives/managers must be able to:
  + Override reservations of regular employees
  + Delete/cancel reservations of regular employees

**9 – Scope of initial release**

* System should manage all conference rooms across all offices (or could we start with a local server or something idk)
* System should schedule and manage video meetings across time zones

**10 – Scope of subsequent release**

* Additional features for scheduling, adding additional roles, etc.
* adding a Front-End

**11 – Stakeholder profiles**

* The sponsors
  + Decide if our project is worth the investment
* Managers
  + Will be interacting with the software regularly and telling us what they want from the software as it’s developed
* IT
  + Must handle bringing the system back up if it goes down
* Legal Team
  + Ensures compliance with local data regulations etc.

**12 – Project Priorities**

* Adding a reservation
* Updating a reservation
* Cancelling a reservation
* Search for open rooms

**13 – Operating Environment**

* Employees will be reserving rooms at any hour of the day
* Employees will be reserving video conferences that occur across different time zones
* Higher management will be cancelling and/or overriding reservations made by regular employees

**14 – Down Time**

* In the event of down time, the following parties will be affected:
* Employees and managers will be unable to interact with rooms or reservations in any way
* Should a meeting need to be scheduled immediately the down time could, for example, cost the company a client due to missing a vital meeting needed to close out a deal

**Use Case Models**

**Request List of Meeting Rooms**:

The system receives a GET and extracts a security token. The Authorization Server is called passing the security token. The Authorization Server credentials response is received. The time requested is compared to existing rooms in the database. The data fields related to rooms open at the location and time requested are displayed to the user.

**Adding New Rooms**

The system receives a POST request and extracts a security token. The Authorization Server is called passing the security token. The authorization credentials are received. Then a new room is added to the database with the data fields from the request.

**Updating Rooms**

The system receives a PUT request and extracts a security token. The Authorization Server is called passing the security token. The authorization credentials are received. If the request is from a Manager, then a room is pulled based on the provided id and updated based on the provided updated field.

**Deleting Rooms**

The system receives a DELETE request and extracts a security token. The Authorization Server is called passing the security token. The authorization credentials are received. If the request is from a manager, then a new room is added to the database with the data fields from the request.

**Request List of Meeting Reservations**:

The system receives a GET and extracts a security token. The Authorization Server is called passing the security token. The Authorization Server credentials response is received. The time requested is compared to existing rooms in the database. The data fields related to room reservations at the location are displayed to the user with the times of reservations.

**Employees can add a meeting reservation.**

The system receives a POST request and extracts a security token. The Authorization Server is called passing the security token. The authorization credentials are received. A database check is performed to see if the room and time conflict with any existing reservations. If there are no conflicts, the reservation is inserted into the database. A successful response is returned to the requester.

Alternate Scenarios

* If attempting to reserve/coordinate a video conference across multiple rooms, the video meetings are reserved in parallel if there are no conflicts.

**Employees can update their own meeting.**

The system receives a PUT request and extracts a security token. The Authorization Server is called passing the security token. The authorization credentials are received. A database check is performed to see if the new requested room and/or time conflict with any existing reservations. If there are no conflicts, the reservation is updated in the database. A successful response is returned to the requester.

**Employees can delete meetings.**

The system receives a DELETE request and extracts a security token. The Authorization Server is called passing the security token. The authorization credentials are received. A database check is performed to see if the reservation exists. If it does not exist, the user will get an error message. If the reservation was booked by the employee who made the request, it will be removed from the database. If the request was made by a manager, it will remove the meeting from the database regardless of who booked it. If an employee requests to delete a meeting that is not their own, they will receive a message declining their request.

**Token Managment**

In the case of no token being passed in, a passed token failing validation, or a token validating but not being authorized for a given request, we will return a failed authorization error to the user. Tokens will be provided to the user upon logging in with the expectation that they will be stored client-side and re-sent when calling other endpoints.