**Don Bosco Institute of Technology**



**MURMUR-A CHATTING APPLICATION**

**Summer Training Project**

**Bachelor in Computer Application**

**Made by.**

**Name: Sambhav chug**

**Roll no.: 00919502022**

**&**

**Name: Rahul Modi**

**Roll no.: 01419502022**

**Guided by: Mr. Amandeep Singh**

**(H.O.D of IT Department)**

**Index**

**1. Introduction**

**2. How we will Develop**

**3. Problem statement**

**4. Why the particular topic is chosen?**

**5. Hard ware and Software required**

* 1. **Hardware**
  2. **Software**
     1. Android Studio
     2. Java
     3. Firebase

**6. The Project Life Cycle**

* + 1. Project Initiation
    2. Execution
    3. Project Completion

**7. Final Review and Evaluation**

**8. User Acceptance Testing (UAT)**

**9. Documentation and Training**

**10. Deployment**

**11. Post-Deployment Support**

**12. Project Closure**

**13. Maintenance and Future Enhancements**

**14. Requirements Analysis Phase**

**15. The Back-end Design Phase**

**16. The Front-end Design Phase**

**17. The Testing Phase**

**18. Theme-Specific Testing**

* + 1. Fear Factor Testing
    2. Immersion Testing
    3. User Experience Testing
    4. Compatibility Testing

**19. The Execution Phase**

**20. The Acceptance Phase**

**21. Conclusion**

**22. Future scope**

**23. Bibliography and reference**

# Introduction:

Murmur is a unique chat application crafted specifically for enthusiasts of the dark and spooky. Unlike generic chat apps, Murmur offers a spooky twist, enhancing the communication experience with a theme that caters to fans of horror and the supernatural. This application enables users to communicate seamlessly over networks, providing functionalities such as displaying messages in a themed chat room, listing online users, and sending personal messages. It’s an ideal medium for horror enthusiasm to stay connected with friends and family in a chilling atmosphere.

The application will:

* Enable a first-time user to register by filling in some personal details,
* Enable a registered user to log on after his/her login details are validated.
* Allow an online user to view a list of other online users.
* Allow users to chat in the common chat room.

The features of our application Murmur are:

Murmur is designed to meet the specific needs of its users with the following features:

1. **User Registration:** First-time users can register by filling in personal details, ensuring a secure and personalized experience.
2. **User Login:** Registered users can log on after their login details are validated, maintaining the security and integrity of the platform.
3. **Online Users List:** Users can view a list of other online users, making it easy to connect and start conversations.
4. **Common Chat Room:** Users can engage in conversations in a common chat room, where the spooky theme enhances the overall experience.

# How we will Develop:

The development of Murmur will be carried out using Java in Android Studio. Java’s platform independence ensures that our chat application can run on various devices, providing flexibility and accessibility. The key concepts utilized in building Murmur include:

* **Chat SDK (Software Development Kit):** Provides essential tools and resources for chat functionality.
* **Messaging Flow:** Ensures smooth and efficient communication between users.
* **Presence:** Tracks user availability and online status.
* **Mobile Push Notification:** Alerts users of new messages and activities.
* **Messaging UI:** Designs the user interface for an intuitive and engaging experience.
* **Chat Server:** Manages the backend operations of the chat system.
* **Offline Support:** Ensures users can access messages and functionalities even without an internet connection.
* **Application Server:** Handles the application’s core functions and services.
* **User Authentication and Manager:** Maintains user security and manages user data.

# Problem Statement:

The market of messaging applications is made for uniform and easy user experience, prioritizing efficiency and functionality over atmosphere and engagement. While existing platforms adequately fulfil basic communication needs, they fail to tap into the growing demand for immersive and emotionally resonant digital experiences. A significant void exists within the market for a messaging app that surpasses the lacking interest, offering users a chilling and suspenseful platform to connect with others who look for the same.

Horror and supernatural themes have captivated audiences for centuries, even Tho it is a very niche community, evoking a sense of fear, wonder, and excitement. Despite the popularity of this genre, the messaging app industry has yet to fully leverage their potential. Current options lack the ability to create a truly terrifying and engaging environment, leaving users yearning for a more immersive and emotionally impactful communication experience.

Ultimately, the problem lies in the lack of a messaging app that effectively combines horror and suspense with a user-friendly interface to deliver a truly terrifying and engaging communication experience.

# Why the particular topic is chosen?

There is a critical opportunity to develop a messaging app that goes beyond mere utility, offering users a platform where they can interact with friends and contacts in a way that is both thrilling and deeply engaging. By combining cutting-edge technology with a deep understanding of horror and suspense, we can create a messaging app that not only meets the basic communication needs of users but also provides a unique and unforgettable experience.

Such a platform has the potential to redefine the messaging application market, attracting a dedicated user base seeking a departure from the ordinary and an escape into a world of fear and excitement. By addressing the limitations of existing messaging apps and capitalizing on the enduring appeal of horror, we can create a groundbreaking product that meets a previously unmet consumer need.

# Hard ware and Software required:

* 1. Hardware
     1. **Android phone**
  2. Software
     1. **Android Studio**
     2. **Firebase**
     3. **Java**

# **5.2.1 Android Studio**

Android Studio is the official integrated development environment (IDE) for Android app development, built and distributed by Google. It includes tools that help software developers design, build, run, and test Android apps.

Android Studio is based on the code editor and developer tools from IntelliJ IDEA and offers additional features to help with productivity. Some of its features include: Compose design tools, Flexible build system, Android Emulator, and Hardware acceleration.

# **5.2.2 Firebase**

Firebase is a set of cloud-based tools that help developers build, deploy, and scale mobile apps. It's a Backend-as-a-Service (BaaS) platform that provides hosted services for a variety of applications, including Android, iOS, JavaScript, Node.js, Java, Unity, PHP, and C++.

Firebase offers products that can help with:

* Building: Get to market quickly and securely with products that can scale globally
* Running: Run your app with confidence and deliver the best experience for your users
* Launching, monitoring, and iterating: Use AI-assistive tools to optimize your app's quality and experience

Firebase provides a robust infrastructure for developers to focus on building the core features of their app, without worrying about managing the backend complexities.

# **5.2.3 Java**

Java is a multiplatform, object-oriented programming language and computing platform that can be used for a variety of applications. It's considered a good first language for learning programming fundamentals and is used by developers to code web applications, mobile apps, enterprise software, big data applications, and server-side technologies. Java is also used in the Internet of Things (IoT) to program smart TVs, cars, heavy machinery, and work facilities.

Java was invented in 1991 by James Gosling of Sun Microsystems and first released in 1995. The language is based on C and C++ and is designed to allow developers to "write once, run anywhere" (WORA), meaning that compiled Java code can run on all Java-compatible platforms without recompilation. Java is also considered fast, secure, and reliable.

Examples where java is used:

* Games: Popular games like Minecraft and RuneScape are written in Java.
* IoT applications: Java can be used to remotely power devices, connect appliances, and more. For example, you can use an app on your phone to turn on your thermostat from a distance.
* Smart cards: Java Card allows secure elements like smart cards to host Java-based applications.

# The Project Life Cycle

The development life cycle of a project usually involves three stages:

* 1. Project Initiation

In this stage, the project plan is prepared, outlining the objectives and outcomes of each stage. The group creates a comprehensive list of tasks, and the project manager assigns responsibilities based on each member’s skills. This stage ensures that all aspects of the project are planned and organized before development begins.

* 1. Project Execution

The execution stage involves several phases to ensure the thorough development and testing of the application:

* + 1. Requirements analysis
    2. Back-end design
    3. Front-end design
    4. Testing
    5. Execution
    6. Acceptance
  1. Project Completion

The project completion stage marks the finalization and delivery of the Murmur chat application. This stage ensures that all project objectives have been met and that the application is ready for public use. It involves several key activities.

# Final Review and Evaluation

The group will conduct a thorough review of the entire project, evaluating each phase of development. This includes revisiting the requirements, design, and testing phases to ensure that all requirements have been completed to the highest standards. The review helps identify any remaining issues or areas for improvement.

# User Acceptance Testing (UAT)

During this phase, selected users, including teacher and potential end-users, are invited to test the application in a real-world environment. Their feedback is crucial for identifying any usability issues or bugs that may not have been caught during the earlier testing phases. This feedback is used to make final adjustments and improvements to the application.

# Documentation and Training

Comprehensive documentation is prepared, detailing the application’s functionality, architecture, and user guides. This documentation serves as a reference for future maintenance and updates. Additionally, training sessions are conducted for the group to ensure they are equipped to assist users and address any issues that may arise.

# Deployment

The final version of Murmur is deployed to the public. This involves setting up the necessary infrastructure, such as servers and databases, to support the application’s launch. The deployment process is carefully managed to ensure a smooth transition from development to live operation.

# Post-Deployment Support

After deployment, the group members provide ongoing support to address any issues that users may encounter. This includes monitoring the application’s performance, fixing bugs, and releasing updates to enhance functionality and security. The support is available to assist users with installation, troubleshooting, and general inquiries.

# Project Closure

The project manager conducts a final project closure meeting with the group to discuss the project’s successes and challenges. This meeting serves as an opportunity to acknowledge the group’s hard work and to document lessons learned for future projects. A project closure report is prepared, summarizing the project’s outcomes, including any deviations from the original plan and how they were addressed.

# Maintenance and Future Enhancements

Although the project is officially completed, the development remains committed to maintaining and improving Murmur. Regular updates are planned to introduce new features, improve performance, and ensure the application remains secure and reliable. User feedback is continuously collected to guide future enhancements and ensure that Murmur continues to meet the needs of its user base.

# Requirements Analysis Phase

During the requirements analysis, we will analyse the requirements that needs to be fulfilled in our chatting application and to identify the problems a meeting with the group member and the teachers is required. To identify the requirements for Chatting Application, we studied the Testing chat utilities at various websites and conducted extensive interviews with chat application users. list of the requirements for the chatting application:

**The application should:**

* Enable a first-time user to register by filling in some personal details
* Enable a registered user to log on after his/her login details are validated
* Allow an online user to view a list of other online users
* Allow users to chat in the common chat room

# The Back-end Design Phase

In this stage, the group decides how the system should function. The formats for data input and output are finalised in this stage. The functional specifications documentation of the system is presented in a language that can be understood by all. The finished project design is, however, executed only on the project manager’s approval.

# The Front-end Design Phase

In this stage, the software components are built. This phase uses the xml file to help in creating the user interface which will be displayed to the user during execution of the program. During the time of development of the software the work is divided among the group members. Some group members were assigned the task of designing the interface of the software.

# The Testing Phase

Software modules are tested for their functionality as per the requirements identified during the requirements analysis phase. To test the functionality of the application, a Quality Assurance (QA) team will be form. The requirements identified during the requirements analysis phase were submitted to the QA team. The QA team test the application for these requirements.

# Theme-Specific Testing - Theme is Horror and Spooky.

* 1. Fear Factor Testing:
     1. Subjective Feedback: Collect qualitative data through user interviews, surveys, and focus groups to gauge emotional responses.
     2. Questions to ask: How did the app make you feel? Were there any specific elements that scared you?
     3. Behavioural Analysis: Observe user behaviour during app usage. Look for signs of fear, such as increased heart rate (if using appropriate technology), physical reactions, or changes in behaviour.
     4. Iterative Refinement: Continuously adjust horror elements based on user feedback. This could involve tweaking sound effects, visuals, or the timing of spooky events.
  2. Immersion Testing:
     1. **Environmental Factors:** Test the app in different lighting conditions and ambient noise levels to assess its impact on immersion.
     2. **Sensory Engagement:** Evaluate the effectiveness of visual, auditory, and potentially tactile elements in creating a spooky atmosphere.
     3. **Consistency:** Ensure that horror elements are consistently applied throughout the app to maintain immersion.
  3. User Experience Testing:
     1. **Balance of Horror and Functionality:** Assess whether the horror elements interfere with core messaging functions.
     2. **User Preferences:** Determine if users can customize the level of horror to suit their comfort level.
     3. **Long-Term Engagement:** Evaluate how the horror theme impacts user retention and engagement over time.
  4. Compatibility Testing:
     1. **Device Compatibility:** Test the app on various devices to ensure that horror elements are displayed and rendered correctly across different screen sizes and resolutions.
     2. **Operating System Compatibility:** Verify that horror features work as intended on different Android versions.

By focusing on these areas, you can ensure that your horror-themed messaging app delivers a truly terrifying and engaging experience for users.

# The Execution Phase

In this phase, based on the pre-define acceptance criteria, the marketing team conducts acceptance testing for the client projects. Acceptance was obtained from the Quality Assurance team. After the project will develop and the people will start using the application, constant support will provide by the team in terms of installation and debugging errors, if any.

# The Acceptance Phase

This phase includes the main publishing of the software and now can be used by public as a daily messaging app which is secure and the only thing which is left is to maintain the software for future use and to secure customers for base on reliability, managing the app for smoother performance and better utilisation of the resources reducing the time to send the message and improving connection with the customers and friends.

# Conclusion

The horror-themed messaging app (Murmur) with its dark and spooky theme, offers a unique messaging experience for horror enthusiasts. The application will be maintained for future use, ensuring reliability and constant improvements in performance and user experience. Murmur aims to create a chilling yet enjoyable communication platform for users who appreciate the eerie and supernatural, making it the go-to app for horror fans to stay connected with friends and family.

The horror-themed messaging app (Murmur) presents a unique opportunity to redefine the communication landscape. By leveraging the power of fear, suspense, and immersive design, this app can create a truly unforgettable user experience. By going beyond the superficial and delving into the psychological aspects of horror, the app can foster a deep emotional connection with users and establish a loyal following.

The app's success will hinge on its ability to maintain a balance between horror and usability. It must be both terrifying and functional, ensuring that users can communicate effectively while being immersed in a chilling atmosphere. Continuous innovation and updates will be essential to keep users engaged and excited about the app's evolving features.

# Future scope

The future of the horror-themed messaging app is filled with exciting possibilities. Some potential areas for expansion include:

* **Adding Community chat:** Its a feature where people connect with their communities in real time and discuss on a similar topic experiencing the same feeling and sharing it with others.
* **Adding share Stories Feature:** It can be a photo, video in short form or any article which you need to share with the people you know will acknowledge it.
* **Augmented reality (AR) integration:** Incorporate AR elements to create interactive horror experiences within the messaging environment.
* **Personalized horror experiences:** Tailor the app's content and atmosphere to individual user preferences.
* **Collaborative storytelling:** Allow users to co-create horror narratives within the app.
* **Horror-themed games and challenges:** Offer interactive elements to enhance user engagement.
* **Merchandise and partnerships:** Explore opportunities for branded merchandise and collaborations with horror-related brands.

By continuously exploring new features and staying ahead of trends, the horror-themed messaging app can solidify its position as a leading innovator in the mobile app industry

# Bibliography and reference

* https://www.geeksforgeeks.org/android-architecture/?ref=lbp
* <https://developer.android.com/studio>
* <https://firebase.google.com/>
* https://firebase.google.com/docs/reference/kotlin/packages