Problem Statement

ในการพัฒนาแอปพลิเคชันแบบ Real-time นั้น การเลือกใช้เทคโนโลยีและเครื่องมือที่ เหมาะสมเป็นสิ่งสำคัญ เพื่อให้สามารถตอบสนองต่อการทำงานแบบออนไลน์ได้อย่างมีประสิทธิภาพ ทีมพัฒนาของเราได้ตัดสินใจใช้ JavaFX สำหรับการพัฒนา Client และ Spring Boot สำหรับการ จัดการ Server

เหตุผลหลักที่เลือกใช้ JavaFX เป็นเครื่องมือสำหรับการพัฒนา Client นั้น นอกจากจะช่วย ให้สามารถสร้างส่วนติดต่อผู้ใช้ (User Interface) ที่ยืดหยุ่นและมีประสิทธิภาพแล้ว ยังเป็นความท้า ทายในการพัฒนาแอปพลิเคชันแบบ Real-time โดยไม่ต้องอาศัย Spring Boot เป็นโครงสร้างหลัก ของ Client การตัดสินใจเช่นนี้ทำให้ทีมพัฒนาต้องออกแบบโครงสร้างของแอปพลิเคชันให้สามารถ ทำงานร่วมกันระหว่าง Client (JavaFX) และ Server (Spring Boot) ได้อย่างมีประสิทธิภาพ

เป้าหมายของโครงการนี้คือการสร้างระบบที่สามารถทำงานได้แบบ Online รองรับการ สื่อสารระหว่าง Client และ Server ได้อย่างรวดเร็วและมีเสถียรภาพ ซึ่งจะเป็นประสบการณ์ที่ท้า ทายและช่วยให้ทีมพัฒนาได้เรียนรู้แนวทางการพัฒนาแอปพลิเคชันแบบ Real-time โดยใช้ Java อย่างเต็มประสิทธิภาพ

Features

คุณลักษณะขั้นต่ำ

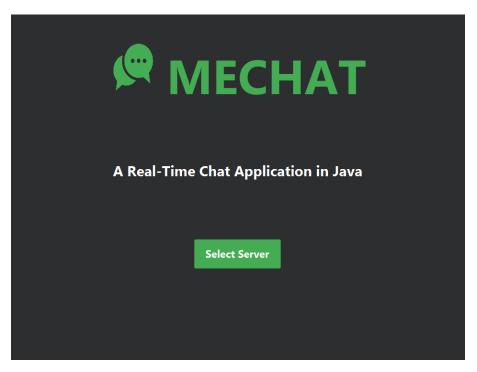
- ผู้ใช้สามารถลงทะเบียนและเข้าสู่ระบบได้
- รองรับการแชทแบบ 1-ต่อ-1 (Private Chat)
- ใช้งานผ่านเครือข่ายอินเทอร์เน็ตได้
- ระบบแจ้งเตือนเมื่อมีข้อความใหม่

คุณลักษณะเพิ่มเติม

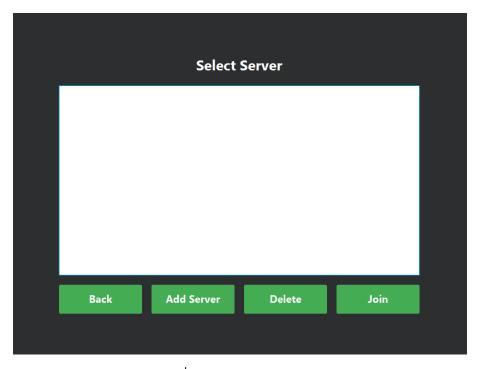
- รองรับการส่งไฟล์และรูปภาพ
- รองรับการแชทเป็นกลุ่ม (Group Chat)
- การเข้ารหัสข้อความเพื่อเพิ่มความปลอดภัย
- ระบบแสดงสถานะออนไลน์/ออฟไลน์ของผู้ใช้
- ระบบจัดการเพื่อน
- แก้ไข/ลบข้อความ

Program Design

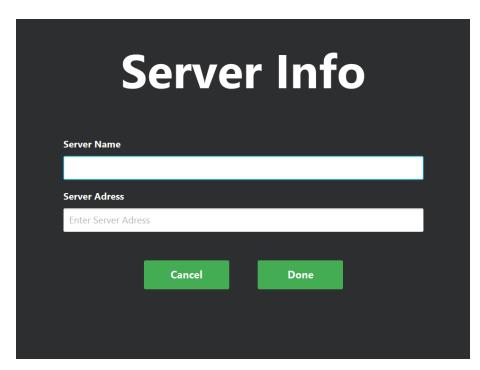
User Interface



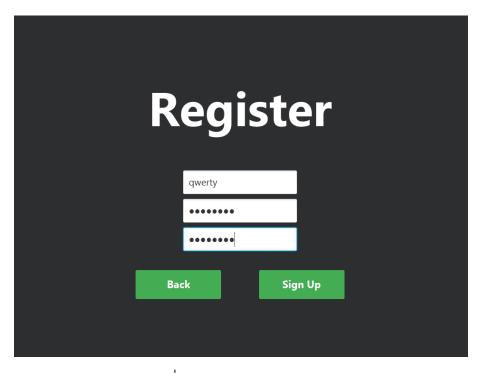
ภาพที่ 1 หน้าต่างแรกเมื่อกดรันโปรแกรม



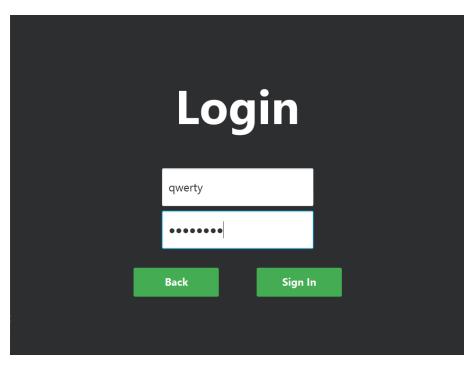
ภาพที่ 2 หน้าต่างเลือก Server



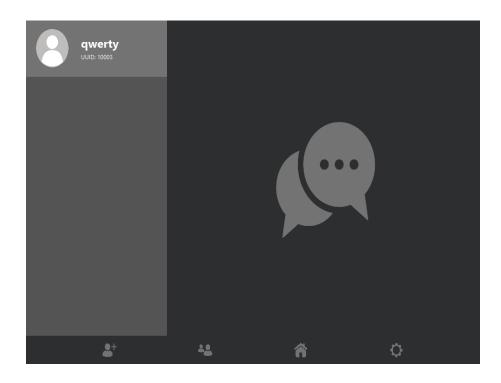
ภาพที่ 3 หน้าต่างกรอกข้อมูล Server



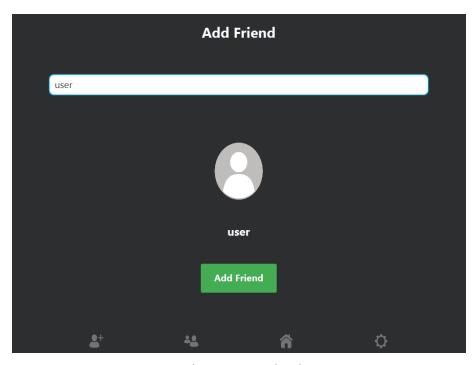
ภาพที่ 4 หน้าต่างการสมัครสมาชิก



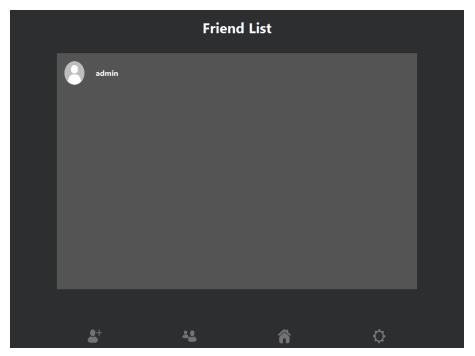
ภาพที่ 5 หน้าต่างการเข้าสู่ระบบ



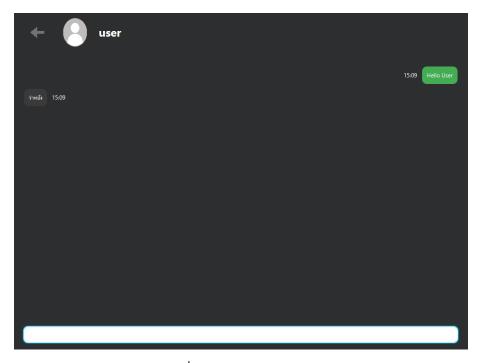
ภาพที่ 6 หน้าหลักของแชท



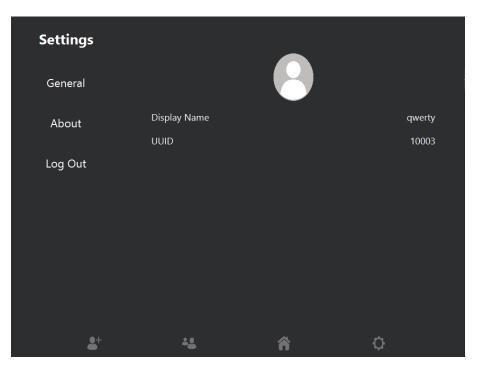
ภาพที่ 7 หน้าต่างเพิ่มเพื่อน



ภาพที่ 8 หน้าต่างรายชื่อเพื่อน



ภาพที่ 9 หน้าต่างแชทสนทนา



ภาพที่ 10 หน้าต่างการตั้งค่าทั่วไป

| Settings | | | | |
|----------|--------------|------------------------------|-----------------------|--|
| | Au | thor: Mechat | Team | |
| General | หากมีข้อสงส่ | เ้ยหรือข้อเสนอแนะ สามารถติดต | เอได้ที่ xxx-xxx-xxxx | |
| About | | | | |
| Log Out | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| ♣+ | 44 | ñ | Φ | |

ภาพที่ 11 หน้าต่างแสดงเกี่ยวกับโปรแกม



ภาพที่ 12 หน้าต่างเมื่อกด Log Out

Diagram

Client

| Арр | Main |
|------------------------|------------------------|
| + main(String[]): void | + main(String[]): void |
| + start(Stage): void | |

Client/App

| ControllerInterface | ViewInterface |
|---------------------|---------------------------|
| + load(): void | + createContent(): Parent |

Client/Interface

| ChatEvent | FriendEvent | MessageEvent |
|--------------------------------|--------------------------------|--------------------------------|
| - log : Logger | - log : Logger | - log : Logger |
| - request : RequestMessage | - request : RequestMessage | - request : RequestMessage |
| - objectMapper : ObjectMapper | - objectMapper : ObjectMapper | - objectMapper : ObjectMapper |
| + handle(RequestMessage); void | + handle(RequestMessage); void | + handle(RequestMessage): void |
| + createChat(int): void | + addFriend(int): void | + sendToPrivateChat(int): void |
| + updateChat(int): void | + updateFriend(int): void | + sendToGroupChat(int): void |

Client/Websocket





| FriendChatInfoController |
|--|
| friendChatinfoView : FriendChatInfoView |
| friendChatinfoModel : FriendChatinfoMode |
| FriandChatInfoController() |
| load(): void |
| backEvent(ActionEvent): void |
| |







| | GroupChatInfoController |
|---------|-------------------------------------|
| - group | ChallnfoView : GroupChallnfoView |
| - group | pChatInfoModel : GroupChatInfoModel |
| + Grou | apChatinfoController() |
| + load | (): void |
| + back | Event(ActionEvent): void |



| AddServerController |
|-----------------------------------|
| - addServerView : AddServerView |
| - addServerModel : AddServerModel |
| + AddServerController() |
| + load(): void |
| + doneEvent(ActionEvent): void |
| cancelEvent/ActionEvent): void |
| |

| | FriendListController |
|---------|------------------------------|
| - frien | dListVlew FriendListVlew |
| - frien | dListModel : FriendListModel |
| + Frie | ndListController() |
| + load | t(): woid |
| + 1080 | tChats(): void |

| SettingCon | rellort |
|--------------------------|--------------|
| - settingView : SettingV | New |
| settingModel : Setting | Model |
| SettingController() | |
| + lead(): void | |
| e generalEvent/Action | Event): void |
| abcutEvent(ActionEv | ent): void |
| + logOutEvent/ActionE | vent): void |

| | HomeController |
|---------------|-----------------------|
| - hameView | : HomeView |
| - hameMod | el : HomeModel |
| - objectMap | per : ObjectMapper |
| - challds : L | ist <string></string> |
| + HomeCor | troller() |
| + load(): voi | d |
| • loadChats | (): void |

LoginController - IsginView : LoginView - IsginModel : LoginModel - objectMapper : ObjectMapper - LoginController() - loginController()



MainControllar
- mainView : MainView
- mainModel : MainModel
+ MainController()
+ load(); void
- serverEvent(ActonEvent); void

Client/ Controller

| AppDataStorage |
|--------------------------------------|
| - appName : String |
| - FILE_PATH : String |
| + saveData(String, String): void |
| + loadData(String); String |
| + deleteData(String): void |
| + createDirectoryIfNotExists(): void |
| + getAppDataPath(): String |

| Time |
|---|
| + getCurrentTime(); String |
| + getFormatTime(String); String |
| + getFormatTime(String, String); String |
| + getFormatDate(String, String, String); String |
| |

| Notification | | | |
|--|--|--|--|
| - log : Logger | | | |
| + showNotification(String, String); void | | | |

| HostWithPortValidator |
|--|
| - IPV4_PATTERN : String |
| - IPV6_PATTERN : String |
| - IPV4_REGEX : Pattern |
| - IPV6_REGEX : Pattern |
| + isValidHostWithPort(String): boolean |

| WindowsSettings | | | | |
|---------------------------|--|--|--|--|
| getWindowsTheme(): String | | | | |

| ScreenHandler | | | | | |
|---|--|--|--|--|--|
| - stage : Stage | | | | | |
| - windowWidth : double | | | | | |
| - windowHeight : double | | | | | |
| - screenHandler : Class extends ScreenHandler | | | | | |
| + ScreenHandler(Stage, double, double) | | | | | |
| + setScreen(ViewInterface); void | | | | | |
| + setScreen(Parent); void | | | | | |
| + debug(Scene); void | | | | | |

| - isFullScreen : boolean |
|---|
| - controllers : ArrayList <controllerinterface></controllerinterface> |
| - server : Map <stringobject,></stringobject,> |
| - authToken : String |
| - session : Session |
| - user : Map <stringobject,></stringobject,> |
| - data : List> <map></map> |
| - chatld : String |
| + MakeCache() |
| + loadControllers(): void |
| + isFullScreen(); boolean |
| + setFullScreen(boolean); void |
| + getController(Class); T |
| + getServer(); Map <stringobject,></stringobject,> |
| + setServer(Map); void |
| + getAuthToken(); String |
| + setAuthToken(String); void |
| + getSession(); Session |
| + setSession(Session); void |
| + getUser(); Map <stringobject,></stringobject,> |
| + setUser(Map); void |
| + getDatas(): List> <map></map> |
| + getData(String); Object |
| + setData(String, Object); void |
| + getChatId(); String |
| |

+ setChatId(String); void

MakeCache

Client/ Utils

RequestMessage

- op : int

- t : int

- d: Map<StringObject, >

- objectMapper : ObjectMapper

+ RequestMessage(String)

+ getOp(): int

+ getT(): int

+ getD(): Map<StringObject, >

ResponseMessage

- op : int

- t : int

- d : Map<StringObject, >

- objectMapper : ObjectMapper

+ ResponseMessage(int, int)

+ getOp(): int

+ getT(): int

+ getD(): Map<StringObject, >

+ put(String, Object): void

+ raw(): String

RestApiService

- log : Logger

- webClient : WebClient

+ connect(): void

+ disconnect(): void

+ getConnection(String, String): Mono<String>

+ login(String, String): Mono<String>

+ register(String, String): Mono<String>

+ getUser(String): Mono<String>

+ getUserByUsername(String): Mono<String>

+ getChat(String): Mono<String>

+ getChatHistory(String): Mono<String>

Client/Service

- usernameProperty : StringProperty - joinButton : Button - searchedFriendName : StringProperty passwordProperty : StringProperty - addServerButton : Button - imagePath : StringProperty - showFriendName : StringProperty - confirmPasswordProperty : StringPro - deleteButton : Button showErrorProperty : StringProperty registerButton : Button - serverListView : ListView<HBox - addFriendButton : Button - backButton : Button - searchField : TextField + RegisterView() + AddFriendView() + createContent(): Parent + createContent(): Parent + createContent(): Parent -friends : ArrayList<Pane> - addGroupMemberButton : Button backButton : Button + createContent(): Parent - friendImageProperty : StringProperty friends : ArrayList<Pane> - userProperty : StringProperty - userProperty : StringProperty uuidProperty : StringProperty - uuidProperty : StringProperty + createContent(): Parent chats : ArrayList<Pane> petFriendImageProperty(); StringProperty
 getBackButton(); Button
 getMessageField(); TextField FriendChatInfoView - aboutButton : Button - logOutButton : Button + createContent(): Parent - editNameButtonLayout : VBox + getChatBox(): VBox + leftBar(): Parent + SettingView() + getUserProperty(): StringProperty + createContent(): Parent + addChat(String, String): void + userInfo(): Parent + getGeneralButton(): Button + getUserProperty(): StringProperty getAboutButton(): Button <u>GroupChatMemberView</u> + getUuidProperty(): StringProperty + getLogOutButton(): Button - serverNameProperty : StringProperty - backButton : Button - friends : ArrayList<Pane> + getChats(): ArrayList<Pane> - serverAddressProperty : StringPropert - showErrorProperty : StringProperty + GroupChatInfoView() + createContent(): Parent doneButton : Button + createContent(); Parent NavbarView - navBar : HBox TemplateView + AddServerView() + createContent(): Parent + createlmageView(String, double, doub # addFriendNavbarButton : Button + createlmageButton(String, double, dou # friendListNavbarButton : Button + navBar(); HBox # homeNavbarButton : Button LoginMenuView + createButtonLayout(String, String); VB usernameProperty : StringProperty - serverButton : Button + createButton(String, String); Button registerButton : Button - passwordProperty : StringProperty + NavbarView() + bindlmage(ImageView, StringProperty) loginButton : Button

LoginMenuView() # setNavbar(): HBox + createContent(): Parent

Client/ View

+ createContent(): Parent

+ getAddFriendNavbarButton(): Button

getFriendListNavbarButton(): Button getHomeNavbarButton(): Button

+ getSettingNavbarButton(): Button

loginButton : Button

+ createContent(): Parent

Server

BaseDTO

- id : Long
- createdAt : LocalDateTime
- updatedAt : LocalDateTime
- + getId(): Long
- + setId(Long): void
- + getCreatedAt(): LocalDateTime
- + setCreatedAt(LocalDateTime): void
- + getUpdatedAt(): LocalDateTime
- + setUpdatedAt(LocalDateTime): void

UserDTO

- username : String
- displayName : String
- avatar : String
- + getUsername(): String
- + setUsername(String): void
- + getDisplayName(): String
- + setDisplayName(String): void
- + getAvatar(): String
- + setAvatar(String): void

Server/Dto

ChatService

- userService : UserService
- chatRepository : ChatRepository
- chatHistoryRepository : ChatHistoryRep
- chatMemberRepository : ChatMemberF
- + getChatsByUserId(Long): List<Chat>
- + getChatById(Long): Chat
- + getChatByUserId(Long, Long): Chat
- + addChat(Chat, List): Chat
- + updateChat(Chat, List, List): void
- + getChatHistory(Chat): List<ChatHistory
- + saveChatHistory(Chat, UserDTO, Strin
- + updateChatHistory(Chat, Long, String)
- + deleteChatHistory(Chat, Long): void
- + getChatUser(Chat, UserDTO): UserDT
- + getChatUsers(Chat): List<ChatMembe
- + getChatUsers(Chat, UserDTO): List<U
- + saveChatUser(Chat, UserDTO): void

FriendService

- friendRepository : FriendRepository
- userService : UserService
- + getFriends(Long): List<Friend>
- + getFriends(Long, Long): Friend
- + addFriend(UserDTO, UserDTO): void
- + updateFriend(UserDTO, UserDTO, Frie

UserService

- userRepository : UserRepository
- + convertToDTO(User): UserDTO
- + convertToEntity(UserDTO): User
- + getAllUsers(): List<UserDTO>
- + getUserById(Long): UserDTO
- + getUserByUsername(String): UserDTC
- + login(String, String): UserDTO
- + createUser(User): User
- + updateUser(Long, User): User
- + deleteUser(Long): void

Server/Service

BoolToIntConverter

- + convertToDatabaseColumn(Boolean): Integer
- + convertToEntityAttribute(Integer): Boolean

- saltRounds : int
- + encrypt(String): String
- + decrypt(String, String): boolean

JWT

- SECRET_KEY : String
- EXPIRATION_TIME : long
- key : SecretKey
- + header(): Header
- + generateToken(Map): String
- + validateToken(String): boolean
- + getPayload(String): Map<StringObject.

YearConverter

- THAI_BUDDHIST_OFFSET : int
- + convertToDatabaseColumn(LocalDateTime): LocalDateTime
- + convertToEntityAttribute(LocalDateTime): LocalDateTime

CustomBanner

- out : PrintStream
- meChatVersion : String
- javaVersion : String
- springFrameworkVersion : String
- springBootVersion : String
- hibernateVersion : String
- mariadbVersion : String
- + printBanner(Environment, Class, PrintStream): void
- + printLogo(): void
- + printCaption(): void
- + print(String, String): void

Server/Utils

BaseEntity

- id : Long
- createdAt : LocalDateTime
- updatedAt : LocalDateTime
- + getId(): Long
- + setId(Long): void
- + getCreatedAt(): LocalDateTime
- + setCreatedAt(LocalDateTime): void
- + getUpdatedAt(): LocalDateTime
- + setUpdatedAt(LocalDateTime): void
- # onCreate(): void
- # onUpdate(): void

Chat

- name : String
- type : Type
- + getName(): String
- + setName(String): void
- + getType(): Type
- + setType(Type): void

ChatAttachments

- chatHistory : ChatHistor
- filePath : String
- + getChatHistory(): ChatHistory
- + setChatHistory(ChatHistory): void

ChatHistory

- chat : Chat
- user : User
- message : String
- edited : Boolean
- deleted : Boolean
- + getChat(): Chat
- + setChat(Chat): void
- + getUser(): User
- + setUser(User): void
- + getMessage(): String
- + setMessage(String): void
- + getEdited(): Boolean
- + setEdited(Boolean): void
- + getDeleted(): Boolean
- + setDeleted(Boolean): void

ChatMember

- chat : Chat
- user : User
- + getChat(): Chat
- + setChat(Chat): void
- + getUser(): User
- + setUser(User): void

Friend

- user : User
- friend : User
- status : Status
- acceptedAt : LocalDateTime
- + getUser(): User
- + setUser(User): void
- + getFriend(): User
- + setFriend(User): void
- + getStatus(): Status + setStatus(Status): void
- + getAcceptedAt(): LocalDateTime

User

- username : String
- password : String
- displayName : String - avatar : String
- + getUsername(): String
- + setUsername(String): void
- + getPassword(): String
- + setPassword(String): void
- + verifyPassword(String): boolean
- + getDisplayName(): String
- + setDisplayName(String): void
- + getAvatar(): String
- + setAvatar(String): void

Server/Entity

Manual

- 1. ติดตั้ง Java JDK 21 จาก <u>Java Archive Downloads Java SE 21</u>
- 2. ติดตั้งฐานข้อมูล <u>MariaDB</u>
- 3. โหลด Source Code จาก Releases · MeChat/256702-F2-Team06-108-175
- 4. ขั้นตอนการติดตั้ง Server
 - 4.1. รันสคริปต์ build.sh หรือ build.bat เพื่อสร้างไฟล์ server.jar ภายในโฟลเดอร์ Build ตัวอย่างการใช้งาน

\$ sh build.sh (Shell Script)

\$ build.bat (Command Prompt)

- 4.2. เปลี่ยนชื่อไฟล์ application.yml.example และ .env.example ให้น้ำ .example ออก
- 4.3. กำหนดข้อมูลใน .env และ application.yml
- 4.4. นำเข้า sql จากไฟล์ Database.sql ลงในฐานข้อมูล
- 4.5. พิมพ์คำสั่ง

\$ java -jar server.jar

- 5. ขั้นตอนการติดตั้ง Client
 - 5.1. รันสคริปต์ build.sh หรือ build.bat เพื่อสร้างไฟล์ client.jar ภายในโฟลเดอร์ Build ตัวอย่างการใช้งาน

\$ sh build.sh (Shell Script)

\$ build.bat (Command Prompt)

5.2. รันสคริปต์ mechat.bat เพื่อเริ่มต้นใช้งาน Client