

РОЗРОБКА ДОДАТКУ ЧАТ-БОТУ ДЛЯ УПРАВЛІННЯ ОПЛАТОЮ ЗА ГУРТОЖИТОК №5 ВНТУ

Вінницький національний технічний університет

Анотація

Розглянуто основні особливості та ключові рішення, прийняті при розробці додатку чат-боту “Dorm5”. Описано головний патерн програмування, який ліг в основу чат-боту, а також використані технології: JavaScript, Node.js, Sequelize, Telegram API.

Ключові слова: чат-бот, JavaScript, Node.js, Sequelize, Telegram API.

Abstract

This article presents the process of building “Dorm5” chat-bot, considering some of the problems we faced and choices we made. The article emphasizes the main software design pattern used in the project and other technologies, such as: JavaScript, Node.js, Sequelize, Telegram API.

Keywords: chat-bot, JavaScript, Node.js, Sequelize, Telegram API.

Вступ

Сьогодні незброєним оком видно, що світ взагалі не такий, як був раніше. Умови карантину, змусили багатьох зовсім по іншому поглянути на світ довкола. Люди уже не можуть уявити життя без смартфона, адже чимало магазинів, сервісів та послуг перейшли в онлайн режим. Тож головним завданням роботи є створення додатку чат-боту для управління оплатою за гуртожиток №5 ВНТУ. Ідеєю такого програмного продукту є надання користувачеві максимально швидкого та зручного доступу до інформації, що стосується його оплати за гуртожиток, наявності заборгованості та просто перевірки балансу або попереднього платежу. Додаток чат-боту розробляється на платформі Telegram, так що у будь-якого типу користувачів не повинно виникати проблем із користуванням. Вибір цієї платформи обґрунтований низькою витратою ресурсів, та тим що цільова аудиторія, це молоді студенти, більша частина з яких уже користується додатком Telegram та навіть є учасником спільноти гуртожитку №5 ВНТУ у додатку Telegram.

Розробка додатку чат-боту «Dorm5»

Dorm5 – це додаток чат-боту для управління оплатою за гуртожиток №5 ВНТУ який працює на платформі додатку Telegram. Під час побудови плану розробки основної частини проєкту виникло питання, який патерн програмування використовувати. Вибір стояв між таки відомими шаблонами, як: MVP, MVVM, та MVC і MVP (Model-View-presenter) – шаблон проєктування, який зазвичай використовується для побудови інтерфейсу користувача. Елемент Model є поданням бази даних, View – реалізує відображення даних з моделі. Частина Presenter у даному випадку є посередником і відповідає за керування подіями з інтерфейсу користувача [2]. В свою чергу, MVVM (Model-View-ViewModel) замість елемента presenter містить ViewModel. Його можна вважати деякою абстракцією, яка містить модель, пов'язану із поданням, і команди, яким подання View може впливати на Model [3]. Остання з розглянутих концепцій MVC (Model-View-Controller) являє собою досить просту та ефективну схему розділення логіки виконання програми, де елемент Controller обробляє та передає запит користувача клієнтові. [4] Було обрано саме цей шаблон проєктування, а як наслідок – JavaScript runtime environment Node.js, що його підтримує. Розглянемо патерн MVC детальніше.

Робота усієї програми базується на взаємодії трьох обов'язкових компонентів: бази даних (Model), контролера (Controller) та подання даних (View). Ключовим вузлом у цій системі є контролер: він опрацьовує вхідні дані і на їх основі регулює роботу з базою даних та з поданням інформації. Простими словами, існує спеціальний механізм, який отримує запит від користувача, потім на його основі робить новий запит уже до бази даних на отримання або зберігання інформаційних ресурсів. База даних виконує потрібні операції та повертає результат, який, знову ж таки, перехоплюється контролером. Отримані дані обробляються та надсилаються до подання, яке відповідає за коректне відображення інформації користувачеві (див. рисунок 1).

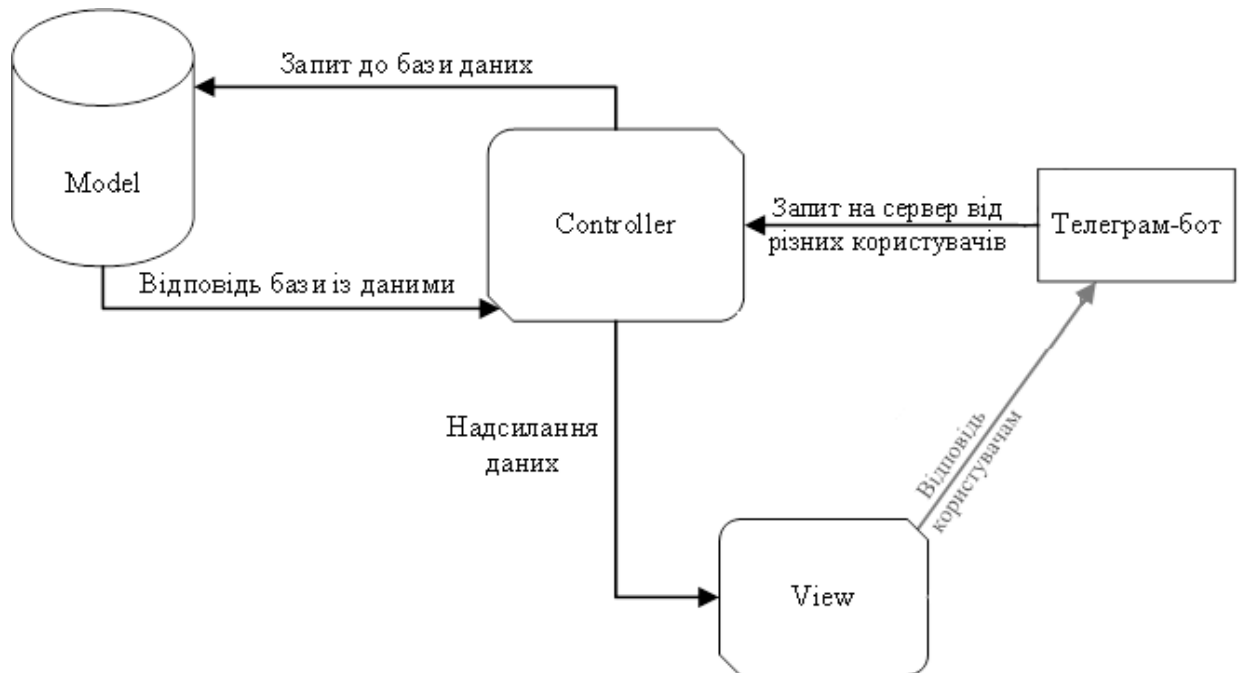


Рис.1. Схема роботи паттерну програмування Model-View-Controller

Використання такого принципу дозволяє чітко розділити роботу над поданням (інтерфейсом програми) і роботу над самою програмою. Також частково вирішується проблема взаємодії з різними типами користувачів, які використовують різні види пристроїв. Можемо інкапсулювати в контролері логіку для обробки запитів як браузера, так і мобільної програми, щоб він в результаті повертав одні і ті ж дані в різному вигляді чи форматі залежно від адресанта. Як наслідок, вся складна робота відбувається в контролері, а на виході маємо чисті перетворені дані. Окрім ізолювання видів логіки, концепція MVC значно зменшує складність великих програм: код виходить структурованим та зрозумілим інтуїтивно, отже і підтримувати та розширювати програму стає набагато легше.

Інші технології, що використовувалися під час розробки

Для розробки серверної частини використовується JavaScript-runtime environment «Node.js» останньої стабільної версії.

Використовується система управління базами даних Sequelize [8]. На етапі розробки всі дані зберігаються на локальному сервері бази даних. Дані зберігаються у вигляді таблиць.

Telegram бот будується з використанням Telegram API [9], що надає сервіс телеграму, а також JavaScript бібліотеки «node-telegram-bot-api». Її особливість у тому, що побудований код є асинхронним, тобто в моменти, коли потік виконання програми мав би заблокуватися, він не блокується, а переходить до виконання іншої задачі.

Висновок

Отже, було розроблено додаток чат-боту «Dogm5», мета якого полягає у реалізації надання користувачеві максимально швидкого та зручного доступу до інформації, що стосується його оплати за куртожиток, наявності заборгованості та просто перевірки балансу або попереднього платежу.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. MVP [Електронний ресурс] Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/Model-view-presenter>
2. MVVM [Електронний ресурс] Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/Model-view-model>
3. MVC [Електронний ресурс] Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/Model-view-controller>
4. Sequelize [Електронний ресурс] Режим доступу до ресурсу: <https://sequelize.org/>
5. Telegram Bot API [Електронний ресурс] Режим доступу до ресурсу: <https://www.npmjs.com/package/node-telegram-bot-api>

Рейда Олександр Миколайович – доцент кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця.

Бондар Vadim Олександрович – студент групи ІПІ-18б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Україна.

Olexandr Reyda – Associate Professor of Software Engineering, Vinnytsia National Technical University, Vinnytsia.

Vadim Bondar – student of group ІPI-18b, Faculty for Information Technologies and Computer Engineering, Vinnytsia National Technical University, Ukraine.