**Location-based Facebook Chatbot**

Table of Contents

[**A.** **Introduction** 5](#_Toc492891098)

[1. Project information 5](#_Toc492891099)

[4. Problem definition 5](#_Toc492891100)

[5. Proposed solution 5](#_Toc492891101)

[5.1. Feature functions 5](#_Toc492891102)

[5.2. Advantages 5](#_Toc492891103)

[5.3. Disadvantages 5](#_Toc492891104)

[6. Functional Requirements 5](#_Toc492891105)

[7. Roles and Responsibilities 5](#_Toc492891106)

[**B.** **Software Project Management Plan** 5](#_Toc492891107)

[1. Problem definition 5](#_Toc492891108)

[1.1. Name of this project 5](#_Toc492891109)

[1.2. Problem abstract 5](#_Toc492891110)

[1.3. Project overview 5](#_Toc492891111)

[1.3.1. Current situation 5](#_Toc492891112)

[1.3.2. The proposed system 5](#_Toc492891113)

[1.3.3. Boundaries of the system 5](#_Toc492891114)

[1.3.4. Future plans 5](#_Toc492891115)

[1.3.5. Development environment 5](#_Toc492891116)

[1.3.5.1. Hardware requirement 5](#_Toc492891117)

[1.3.5.2. Software requirement 5](#_Toc492891118)

[2. Project organization 6](#_Toc492891119)

[2.1. Software process model 6](#_Toc492891120)

[2.2. Roles and responsibilities 6](#_Toc492891121)

[2.3. Tools and techniques 6](#_Toc492891122)

[2.4. Dependencies 6](#_Toc492891123)

[3. Project Management plan 6](#_Toc492891124)

[3.1. Tasks 6](#_Toc492891125)

[3.1.1. Initializing 6](#_Toc492891126)

[3.1.2. Planning 6](#_Toc492891127)

[3.1.3. Specifying requirements 6](#_Toc492891128)

[3.1.4. Designing database 6](#_Toc492891129)

[3.1.5. Create software design documents 6](#_Toc492891130)

[3.1.6. Implementing 6](#_Toc492891131)

[3.1.7. Testing 6](#_Toc492891132)

[3.1.8. Creating user’s manual 6](#_Toc492891133)

[3.2. Tasks sheet 6](#_Toc492891134)

[3.3. All meeting minutes 6](#_Toc492891135)

[4. Coding convention 6](#_Toc492891136)

[**C.** **Software Requirement Specification** 6](#_Toc492891137)

[1. User requirement specification 6](#_Toc492891138)

[1.1. Guest requirement 6](#_Toc492891139)

[1.2. User requirement 6](#_Toc492891140)

[1.3. Administrator requirement 6](#_Toc492891141)

[2. System requirement specification 6](#_Toc492891142)

[2.1. External interface requirement 6](#_Toc492891143)

[2.1.1. User interface 6](#_Toc492891144)

[2.1.2. Hardware interface 6](#_Toc492891145)

[2.1.3. Software interface 6](#_Toc492891146)

[2.1.4. Communication protocol 6](#_Toc492891147)

[2.2. System overview use case 6](#_Toc492891148)

[2.3. Use cases 6](#_Toc492891149)

[3. Software system attribute 6](#_Toc492891150)

[3.1. Usability 6](#_Toc492891151)

[3.1.1. Graphic user interface 6](#_Toc492891152)

[3.1.2. Usability 6](#_Toc492891153)

[3.1.3. Installation 6](#_Toc492891154)

[3.2. Reliability 6](#_Toc492891155)

[3.3. Availability 6](#_Toc492891156)

[3.4. Security 7](#_Toc492891157)

[3.5. Maintainability 7](#_Toc492891158)

[3.6. Portability 7](#_Toc492891159)

[3.7. Performance 7](#_Toc492891160)

[4. Conceptual program 7](#_Toc492891161)

[**D.** **Software Design Description** 7](#_Toc492891162)

[1. Design overview 7](#_Toc492891163)

[2. System architecture design 7](#_Toc492891164)

[2.1. Web application architecture description 7](#_Toc492891165)

[3. Component diagram 7](#_Toc492891166)

[4. Detailed description 7](#_Toc492891167)

[4.1. Class diagram 7](#_Toc492891168)

[4.2. Class diagram explanation 7](#_Toc492891169)

[4.3. Interactive diagram 7](#_Toc492891170)

[5. User interface design 7](#_Toc492891171)

[6. Database design 7](#_Toc492891172)

[6.1. Entity relationship diagram 7](#_Toc492891173)

[6.2. Entity dictionary 7](#_Toc492891174)

[6.2.1. AspNetUsers 8](#_Toc492891175)

[6.2.2. AspNetUserClaims 8](#_Toc492891176)

[6.2.3. AspNetUserLogins 8](#_Toc492891177)

[6.2.4. AspNetUserRoles 8](#_Toc492891178)

[6.2.5. AspNetRoles 8](#_Toc492891179)

[6.2.6. Campaign 8](#_Toc492891180)

[6.2.7. Combo 8](#_Toc492891181)

[6.2.8. ComboProductMap 8](#_Toc492891182)

[6.2.9. Corporation 8](#_Toc492891183)

[6.2.10. Coupon 8](#_Toc492891184)

[6.2.11. ImageLink 8](#_Toc492891185)

[6.2.12. Order 8](#_Toc492891186)

[6.2.13. OrderDetail 8](#_Toc492891187)

[6.2.14. Product 8](#_Toc492891188)

[6.2.15. ProductCategory 8](#_Toc492891189)

[6.2.16. ProductCatgoryMap 8](#_Toc492891190)

[6.2.17. Promotion 8](#_Toc492891191)

[6.2.18. Shop 8](#_Toc492891192)

[6.2.19. ShopCategory 8](#_Toc492891193)

[6.2.20. ShopCategoryMap 8](#_Toc492891194)

[7. Algorithms 8](#_Toc492891195)

1. **Introduction**
   1. Project information
      * Project name: **Location-based Facebook Chat bot**
      * Project code: **LBFC**
      * Product type: **Facebook messenger bot + Web application**
      * Start date: **September 4th, 2017**
      * End date: **December 12th, 2017**
   2. Introduction

In this document, we will introduce the Location-based Facebook Chat bot solution, which helps customers shopping easier without any required app but Facebook messenger – an app that every Facebook user need to install. Besides, it can help shops admin manage products and other services such as promotions or coupons and orders.

* 1. Current situation

When a shop or a company want to promote their products and services, a mobile app is needed since a web page on smartphone can’t actually replace all the functions in a desktop web page better than a mobile app. However, many companies are now creating their own apps. This can make users feel uncomfortable since they have to install an app whenever they visit a new store, which can make their phones slow down.

* 1. Problem definition

Current problems:

* **Too many steps to install an app**: To install an app, users have to process a lot of steps to get app installed. On the other hand, more apps installed means less storage available on the phone. Therefore, people prefer less app but with same functionality.
* **Website for mobile can’t display all functions**: Because of the small screen, a website for mobile needs to reduce all details to leave more space for primary functions. This lead to lack of functionality. If the webpage decides not to reduce then the user experience will become badly.
  1. Proposed solution

Our solution is Location-based Facebook Chat bot. This bot will help customers explore shops just by chatting. From product information to promotion information and ordering, this bot can automatically do that without any click. Furthermore, it can detect customer location via Wi-Fi and recommend the services in the shop and surrounding stores. Customer can also get coupon or promotion from this bot.

* + 1. Feature functions
* **Auto-detect customer location:** When a customer visits a shop, connect to Wi-Fi or enter a code from his/her invoices, system automatically detects where the customer is and process to another functions based on the location.
* **Recommend products and promotions:** Based on the location, system will search for products of the current shop and surrounding shops, also the promotions.
* **Allow customers to order:** customers can order products in the current shop or nearby shops. System will return an order code for each customer to track their orders.
* **Real-time notifications:** notifications about services of the current shop and nearby shops will be send in real-time. When a shop update its services, users will be notified immediately.
* **Service management:** Staffs from all the shop associated with system can control the orders and services via one site only.
  + 1. Advantages
* **Simple to use:** customers don’t need to install any app to use this system. All customers need is to install Facebook messenger, which is really popular since it is another must-installed for Facebook user.
* **Customers don’t have to research for places:** information of shops are all included in the system, therefore users don’t have to search around to get information about services of shops nearby. Besides, the system is in real-time so customers don’t have to refresh anything and the bot will automatically advice new services.
* **Customers can get coupon on their own phone:** Usually customers have to come to a store to receive coupon. In this system, a coupon is presented as code and users only have to copy the code for the bot. Then, coupon can be used.
* + 1. Disadvantages
* **A bot can’t replace a human:** An AI bot can’t replace a human totally and it still need some supporters in case of strange behaviors from customers.
* **Some business won’t suit for all brands:** because the management site is created for all brands, it can’t meet some special requirements of some brands.
  1. Functional Requirements
     1. For guests
* Order:
  + See full list of menu of a shop
  + Create order
  + See full list of order detail
  + Cancel order
* Promotion and Coupons:
  + Get coupons and promotions
  + Apply coupons and promotions in order
  + See list of available coupons and promotions
* Search for nearby shops
  + 1. For shop staffs
* Manage products
  + Add product to inventory
  + Update product information
  + Get product information
  + Delete product
* Manage promotions
  + Add promotions
  + Get promotions information
  + Update promotions information
  + Delete promotion
  + Deactivate promotions
* Manage orders
  + Get all orders information in shop
  1. Roles and Responsibilities

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Name | Role | Position | Contact |
| 1. | Tường Đức Khang | Developer | Leader | [khangtdse61752@fpt.edu.vn](mailto:khangtdse61752@fpt.edu.vn) |
| 2. | Thái Hoàng Duy Bảo | Developer | Member | baothdse |
| 3 | Nguyễn Văn Hùng | Developer | Member | hungnvse61305@fpt.edu.vn |
| 4. | Dương Hải Quang | Developer | Member | quangdhse61383@fpt.edu.vn |

1. **Software Project Management Plan**
   1. Problem definition
      1. Name of this project

Official name: Location-Based Facebook ChatBot

Vietnamese name: Hệ thống ChatBot theo địa điểm

Abbreviation: LBFC

* + 1. Problem abstract

To support users to find the location where can help them to see products, promotion and service. We concentrate on the one thing familiar with people nowadays - their smart phones. We provide a chatbot which can help users detect the shop they are staying. After that, they easily to find the products, promotion they need .In other words, a chatbot help you to order product easily.

Moreover, we develop a website as alternative using way for manage product, store system. Generally, It’s also provide user information about product, price how to get the coupons and payment.

* + 1. Project overview
       1. Current situation

Some problems encountered in this project:

* User’s behavior: User use teen code , regional languages or Vietnamese language that make chatbot can’t understand all of situation.
* Variety of users: People have different favorite tastes. The work of finding beverage and food to satisfy all of them is very difficult.
* Gathering data: Websites of different store have different interfaces and structures, which makes gathering data more difficult.
* Process of chatbot: People can send the picture to chatbot. So chatbot hardly to understand to response user. Moreover, chatbot follow the story step by step.
  + - 1. The proposed system

The proposed system is an application that can help customer find product in place thereabouts. Moreover, the system suggests any advice and product and deal to customer. That make the customer have more choice and easily to their decision.

Separated based on user’s role, the system offers the following functions:

* For guest:
* Search products.
* View product’s information, promotion, services.
* Register member.
* Order product and cancel booking.
* Chat with chat bot.
* Get coupons on chat bot interface.
  + - 1. Boundaries of the system
* The language of system is Vietnamese.
* Allow user to interact with Facebook user via messages and comments only from the time Facebook page connected to the system.
* Setting for luis.ai is only limit to expressions of intents.
* No mobile applications.
* Allow user to manage products, categories, orders of a shop.
  + - 1. Future plans

In the future, we want to upgrade and develop the system following features:

* Allow chat bot can understand any language when user chatting.
* Allow user order whatever they want (electronic device, smart phone…)
* Allow bot can understand clearly any case of customer and any acronyms.
* Allow Admin to change setting for api.ai not only intents but also entities, context, training.
  + - 1. Development environment
         1. Hardware requirement
         2. Software requirement
  1. Project organization
     1. Software process model
     2. Roles and responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Full name** | **Role in Group** | **Responsibilities** |
| 1 | Lâm Hữu Khánh Phương | Supervisor,  Product owner | * Specify user requirements * Control the development process * Support in technique and business logic. |
| 2. | Tường Đức Khang | Team Leader,  BA,  Developer,  Tester | * Create system architecture design. * Support technique * Arrange daily meeting * Clarify requirements * Design database. * Design GUI * Code * Write document and report * Create test plan. * Test |
| 3. | Thái Hoàng Duy Bảo | Team member,  BA,  Developer,  Tester | * Clarify requirements * Design database. * Design GUI * Code * Write document and report * Create test plan. * Test |
| 4 | Nguyễn Văn Hùng | Team member,  BA,  Developer,  Tester | * Clarify requirements * Design database. * Design GUI * Code * Write document and report * Create test plan. * Test |
| 5 | Dương Hải Quang | Team member,  BA,  Developer,  Tester | * Clarify requirements * Design database. * Design GUI * Code * Write document and report * Create test plan. * Test |

* + 1. Tools and techniques
       1. For Development
* Font-end technologies:
* HTML5
* CSS, Bootstrap
* Javascript
* Jquery
* Back-end:
* Website: ASP.NET,MVC5
* Web server: Azure
* Database management system: SQL Server 2014
  + - 1. For Management
* Task tracking: Microsoft Excel, Github, Trello
* Storing and managing source code : Github, Googledrive, Source tree
  + - 1. For Communication
* Facebook
* E-mail
* Skype
  + 1. Dependencies
  1. Project Management plan
     1. Tasks
        1. Initializing
        2. Planning
        3. Specifying requirements
        4. Designing database
        5. Create software design documents
        6. Implementing
        7. Testing
        8. Creating user’s manual
     2. Tasks sheet
     3. All meeting minutes

All Meeting Minutes are put in folder “Meeting minutes” in the attached CD.

* 1. Coding convention
     1. Naming Convention
* In short examples that do not include using directives, use namespace qualifications. If you know that a namespace is imported by default in a project, you do not have to fully qualify the names from that namespace. Qualified names can be broken after a dot (.) if they are too long for a single line, as shown in the following example.
* You do not have to change the names of objects that were created by using the Visual Studio designer tools to make them fit other guidelines.
  + 1. Layout Convention
* Use the default Code Editor Settings (smart indenting, four-character indents, tabs saved as spaces).
* Write only one statement/declaration per line.
* If continuation lines are not indented automatically, indent them one tab stop (four spaces).
* Add at least one blank line between method definitions and property definitions.
* Use parentheses to make clauses in an expression apparent.
  + 1. Commenting Convention
* Place the comment on a separate line, not at the end of a line of code.
* Begin comment text with an uppercase letter.
* End comment text with a period
* Insert one space between the comment delimiter (//) and the comment text.
  + 1. Language Guidlines

<https://msdn.microsoft.com/en-us/library/vstudio/ff926074.aspx>

1. **Software Requirement Specification**
   1. User requirement specification
      1. Guest requirement
      2. User requirement
      3. Administrator requirement
   2. System requirement specification
      1. External interface requirement
         1. User interface
         2. Hardware interface
         3. Software interface
         4. Communication protocol
      2. System overview use case
      3. Use cases
   3. Software system attribute
      1. Usability
         1. Graphic user interface
         2. Usability
         3. Installation
      2. Reliability
      3. Availability
      4. Security
      5. Maintainability
      6. Portability
      7. Performance
   4. Conceptual program
2. **Software Design Description**
   1. Design overview
   2. System architecture design
      1. Web application architecture description
   3. Component diagram
   4. Detailed description
      1. Class diagram
      2. Class diagram explanation
      3. Interactive diagram
   5. User interface design
   6. Database design
      1. Entity relationship diagram
      2. Entity dictionary

|  |  |
| --- | --- |
| Entity name | Description |
| AspNetUsers | Storing user information. User can be admin or shop staffs, not guest. |
| AspNetUserClaims |  |
| AspNetUserLogins | Storing information of 3rd party login provider, such as Google, Facebook. |
| AspNetUserRoles | Mapping between users and roles |
| AspNetRoles | Storing roles |
| Campaign | Represent a promotion campaign |
| Combo | Represent a combo of products |
| ComboProductMap | Mapping between combo and product, tells which product is in which combo |
| Corporation | Represent a brand |
| Coupon | Storing coupons information |
| ImageLink | Storing image link for product, shop. |
| Order | Storing order |
| OrderDetail | Storing details of orders |
| Product | Represent a product |
| ProductCategory | Represent categories of products |
| ProductCatgoryMap | Mapping between products and products categories |
| Promotion | Represent a promotion |
| Shop | Represent a shop |
| ShopCategory | Represent a shop category |
| ShopCategoryMap | Mapping between shop and category |

* + - 1. AspNetUsers
      2. AspNetUserClaims
      3. AspNetUserLogins
      4. AspNetUserRoles
      5. AspNetRoles
      6. Campaign
      7. Combo
      8. ComboProductMap
      9. Corporation
      10. Coupon
      11. ImageLink
      12. Order
      13. OrderDetail
      14. Product
      15. ProductCategory
      16. ProductCatgoryMap
      17. Promotion
      18. Shop
      19. ShopCategory
      20. ShopCategoryMap
  1. Algorithms