WANBO : Developing Warnet Booking Application with FDD Method

Daniel Christopher Kesuma   
Computer Science Department  
School of Computer Science

Bina Nusantara UniversityJakarta, Indonesia 11480  
[daniel.kesuma@binus.ac.id](mailto:daniel.kesuma@binus.ac.id)

Levandio Yuvens Jonathan  
Computer Science Department  
School of Computer Science

Bina Nusantara UniversityJakarta, Indonesia 11480  
[levandio.jonathan@binus.ac.id](mailto:levandio.jonathan@binus.ac.id)

Moch Kelvin RA  
Computer Science Department  
School of Computer Science

Bina Nusantara UniversityJakarta, Indonesia 11480  
[moch.kelvin@binus.ac.id](mailto:moch.kelvin@binus.ac.id)

*Abstract*— This research was conducted to develop an online booking application that focuses on internet cafes. This application was made because there isn’t any application reservation for internet cafes, where the problem is that we have to contact the admin if it is empty or full. In developing this application, we use the Feature Driven Development methodology because it can facilitate the development of applications that focus on its features. This can be seen because the plan is that this application will have several features that make it easier for users to use it and also make it easier and more efficient for developers to develop it.

Keywords— Feature Driven Development (FDD), Internet Cafe, Agile

# Introduction

An internet cafe or cybercafe is a place where one can use a computer with Internet access, most for a fee, usually per hour or minute; sometimes one can have unmetered access with a pass for a day or month, etc. It may serve as a regular café as well, with food and drinks being served. Ideally, Internet cafes in developing countries represent reasonably priced access points to sources of information for personal development, business start-up and growth, or political participation and the progress of civil society. Indonesia, as one of the developing countries, has around 13000 Internet cafes based on Badan Pusat Statistik (BPS) data.

In the traditional booking system, a customer has to make a phone call in order to book a room / pc. If luckily the phone gets connected, then the customer does some conversation with the owner or operator in the internet cafe. Then he demands for the internet cafe’s package and do some discussion over the package then he orders and he has to give some of this identification specifications. This process takes 3-4 minutes to complete. On the receiver side there is hardy one phone line and one operator. so he can cover around 15-20 orders maximum in an hour.

There are lots of areas to be solved for current internet cafes using the modern IT world. Many areas come like human resource management, accounts management, etc. We can improve those areas of the traditional booking system with an online booking system, so booking process will become faster, more efficient, and easier to manage and report. These advantages will lead to better productivity and higher profit for the internet cafe. That’s why we decided to make Wanbo (Warung Internet Booking Online) so it’s easier to reserve rooms or PCs and make orders in an Internet cafe. Customers can find out when the cafe is full and can order at certain hours when the cafe is empty.

The application’s concept is similar to a well-known cinema ticket booking application. The difference is, Wanbo is used to book rooms or PCs in an internet cafe instead of seats in a cinema theatre. Wanbo will cooperate with a determined Internet cafe to help users book rooms or PCs and order foods or drinks in that place from the application and help the internet cafe’s administrator to manage the booking and order easily and also monitor the transactions with the report analysis that Wanbo made.

To develop this application, we are going to use one of the agile method in the Software Development Life Cycle (SDLC) that is Feature Driven Development (FDD). The FDD model was chosen because this model focuses on the features so that each feature can be developed optimally and make sure the functionality objective of the application is achieved. FDD will start with developing the overall model, then building a features list. After that, the plan by feature phase begins. Lastly, the iterative processes of design by feature and build by feature are the two final phases in FDD.

# Methodology

The method used in Wanbo development is the FDD method. Based on Palmer (2001). The FDD method is a stage that is designed and implemented to produce repetitive work results within a certain time frame and is easy to measure. FDD helps engineers to have substantial working outcomes. It highlights some user esteemed values and sorts those values into business-related capabilities. FDD encourages the group of engineers to produce outcomes at regular intervals and offers advance following and announcing abilities..

The following are the stages in the FDD method:

## Stage #1: Build an Overall Model.

The first step in FDD is making a detailed model of the system. Using the primary goal of the application as a guide, detailed domain models that will be merged into an overall model are developed. This overall model will act as an outline and draft in the next development processes.

## Stage #2: Build a Feature List.

Using the outline that has been created in the first step, the engineer will make a list of customer’s required features that propose a specific goal.

## Stage #3: Plan by Features.

In this step, the team will examine some factors like complexity or user value of a feature to determine the order in which each feature will implement, as well as the team members that will be assigned to each feature set.

## Stage #4: Design by Feature.

All plans that have been made and analyzed in stage 3 will be carried out sequentially according to their level of importance. The rough design that has been made in the previous stage will be more detailed, starting from the mockup design, program flow, and others.

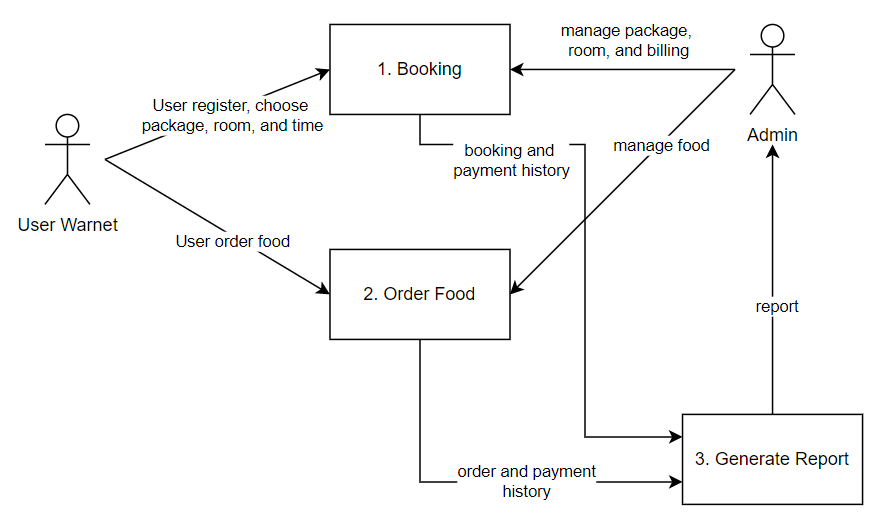
## Stage #5: Build by Feature.

This step implements all the models and plans that have been developed in the previous steps. Here, user interfaces are built, as are components detailed in the technical design, and a feature prototype is created. The unit is tested, inspected, and approved, then the completed feature can be promoted to the main build.

# Implementation

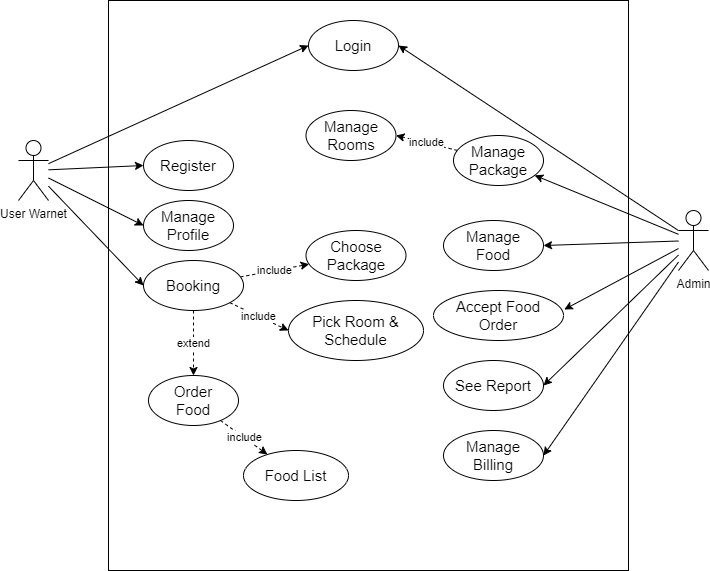
## Build an Overall Model

All requirements needed will be gathered in this step. The actor in this model application is defined, so it will make model sketching easier .Here is the overall model of Wanbo:

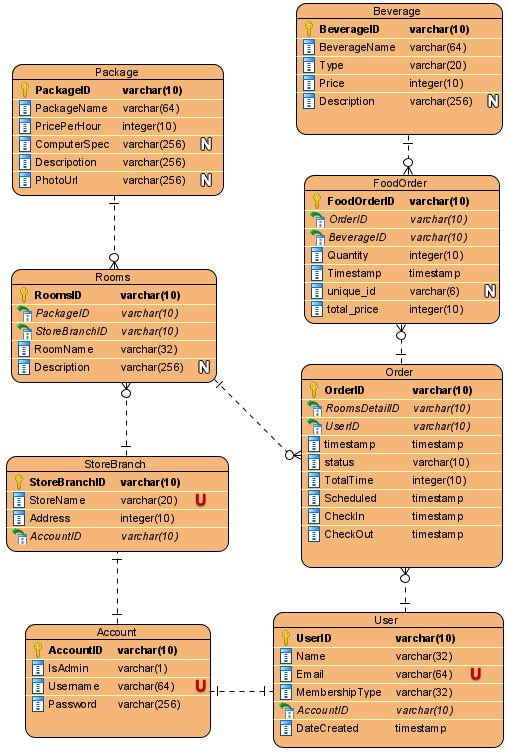


1. Overall system flow

After getting the big picture of the app, a use case diagram and entity relationship diagram (ERD) will be created to understand all features that will be developed.

****

1. Use case diagram



1. Entity relationship diagram

## Build a Feature List

Referring to the ERD and Use cases that have been made in the previous stage, all feature lists can now be listed in detail. Here is the overall model of Wanbo:

1. User Feature List

|  |  |  |
| --- | --- | --- |
| **No.** | **Feature** | **Detail** |
| 1. | Register | - |
| 2. | Login | - |
| 3.  4.  5.  6. | Dashboard | See package info  See billing info  See food menu  See billing time left |
| 7.  8.  9. | Profile | Edit profile  Logout  See booking history |
| 10  11.  12.  13.  14.  15.  16. | Booking | Create new booking  Choose package  Choose room  Choose time  Pay  Reschedule book  Check in |
| 17.  18.  19. | Order Food | See menu  Choose menu  Pay |

1. Admin Feature List

|  |  |  |
| --- | --- | --- |
| **No.** | **Feature** | **Detail** |
| 1. | Login / Logout | - |
| 2. | Edit Profile | - |
| 3.  4.  5.  6.  7.  8.  9.  10. | Manage Package | See package list  Insert new package  Update package  Delete package  See room list  Insert new room  Update room  Delete room |
| 11.  12.  13.  14.  15.  16. | Manage Food | Show food data  Insert new food data  Update food data  Delete food data Show Food order  Accept food order |
| 17.  18.  19. | Manage Billing | See active billing  Add billing time  Block billing |
| 20.  21.  22. | See Report | See payment history  See order history  See reports within a certain time period |

## Plan by Feature

All the features from the previous step will be analyzed and assigned a priority which determines which feature will be developed first. In this planning process, we are using a simple timetable that states the development duration of each feature.

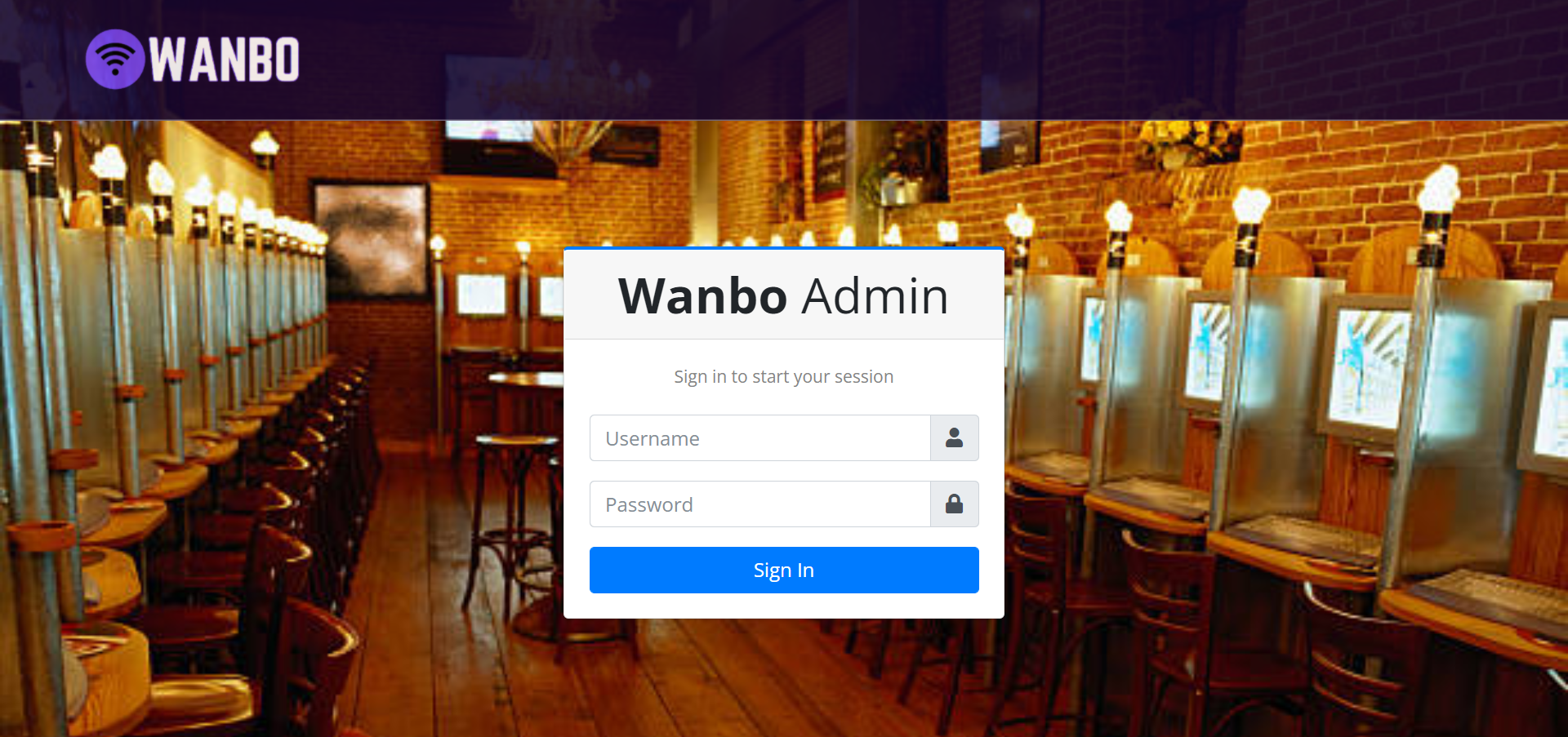
1. Plan by Feature

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **User level** | **Name** | **Duration**  **(days)** |
| 1. | - | Preparing database | 3 |
| 2. | Admin | Preparing base application | 1 |
| 3. | Admin | Login/logout | 1 |
| 4. | Admin | Manage billing | 3 |
| 5. | Admin | See report | 4 |
| 6. | Admin | Manage package | 2 |
| 7. | Admin | Manage food | 1 |
| 8. | Admin | Edit profile | 1 |
| 9. | User | Preparing base application | 1 |
| 10. | User | Register | 1 |
| 11. | User | Login/logout | 1 |
| 12. | User | User dashboard | 3 |
| 13. | User | Booking | 4 |
| 14. | User | Order food | 2 |
| 15. | User | Profile | 2 |
| Total | | | 30 |

The development process is scheduled to be done in one month. After the development is complete, it will take another month to test and fix the error that was found.

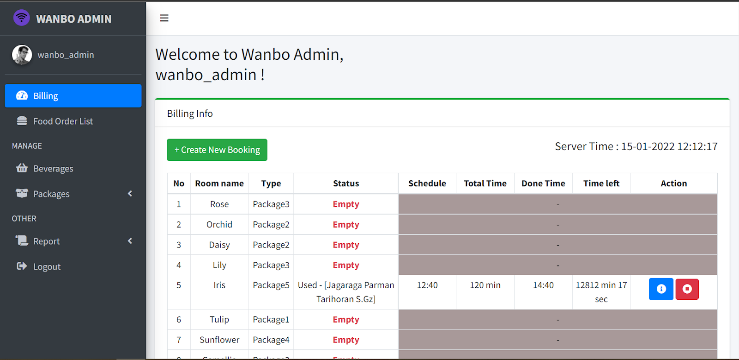
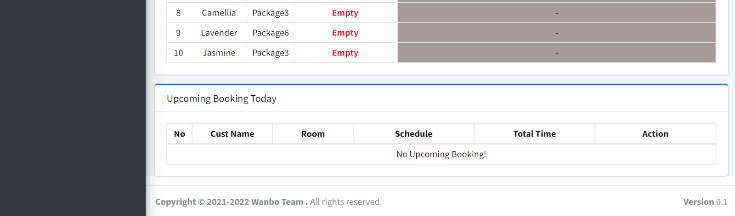
## Design by Feature

After listing and planning the features, the next step is designing the display of each feature so that it is easier to develop and review.



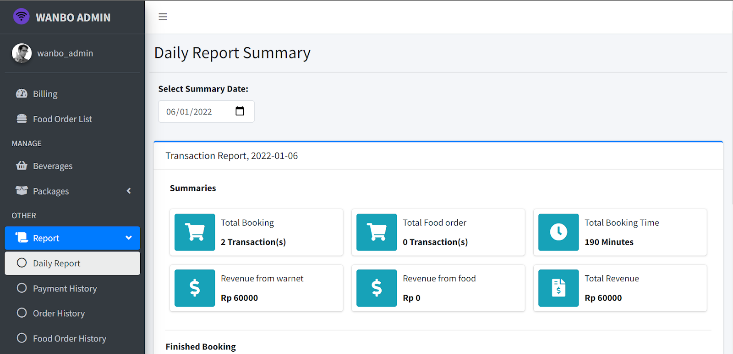
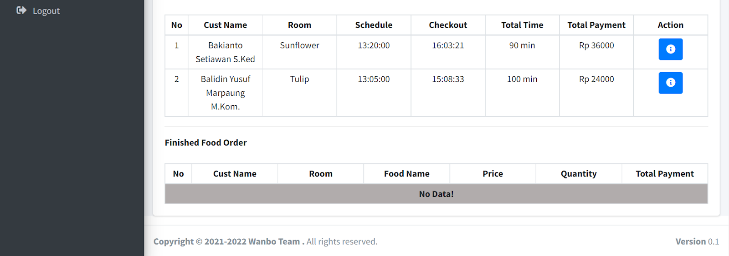
1. Admin login

This is the login page design for Wanbo admin. Admin need to fill his username and password to login into the application.



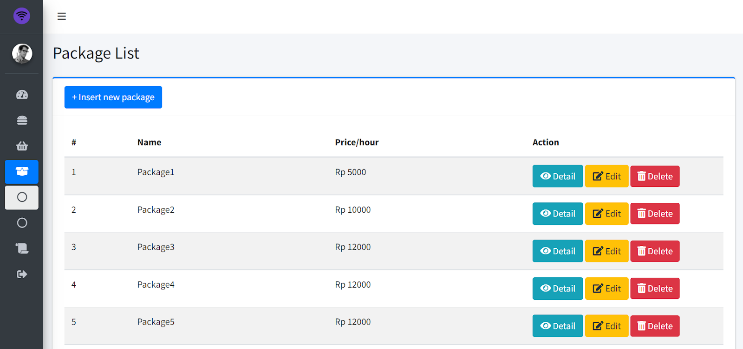
1. Admin dashborad

This is the admin dashboard design where the admin can see and manage the ongoing and upcoming billing.



1. Admin report page

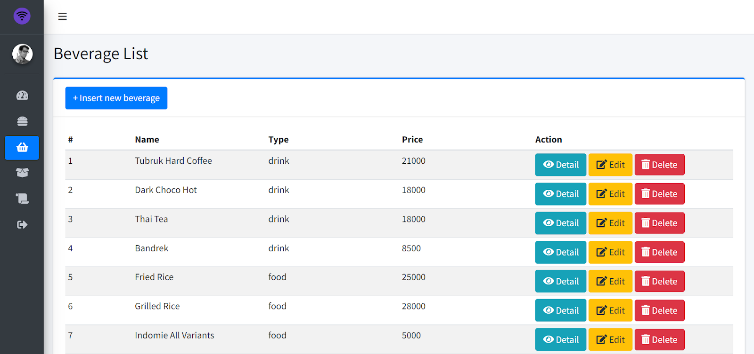
In this page, admin can see daily report summary from total transaction, booking transaction, food order transaction, also total revenue. Beside this summary page, each type of transaction also has its own history page like food order history, payment history, and order history.



1. Manage packages

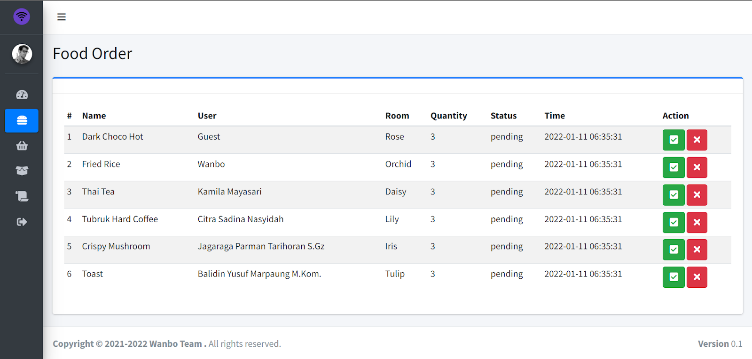


1. Manage rooms



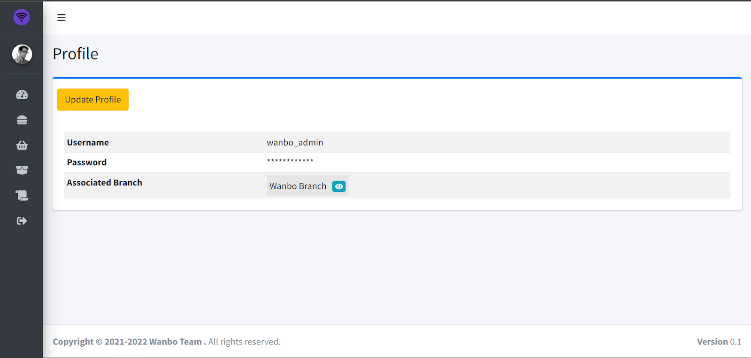
1. Manage beverages

In the manage menu is designed so that the admin can show, create, update, and delete beverages, packages, and rooms easily.



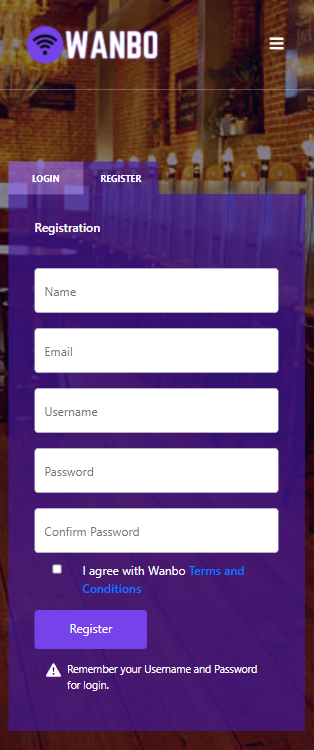
1. Food order list

Beside seeing and managing billing, the dashboard is also designed with food order list that make admin can see, accept or decline incoming food orders easily.

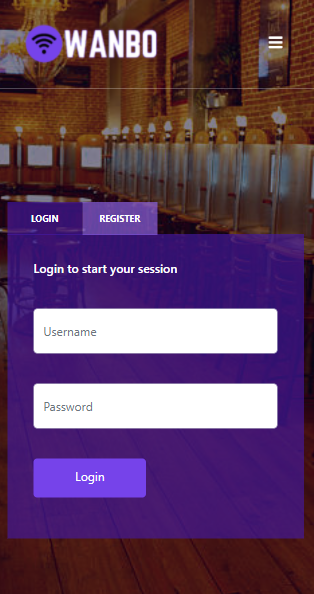


1. Admin profile

This is the admin profile design where admin can see his profile and the branch that he manages. Admin can also edit his profile on this page.

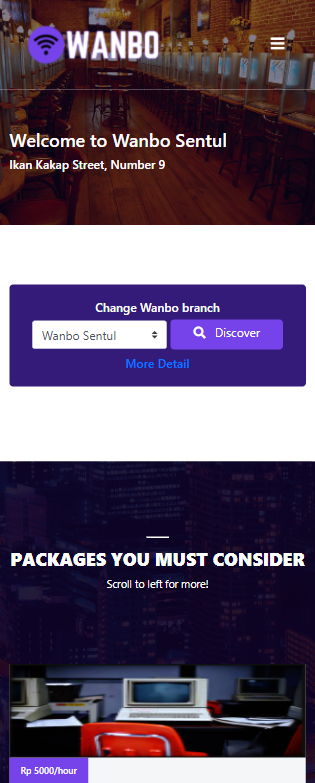
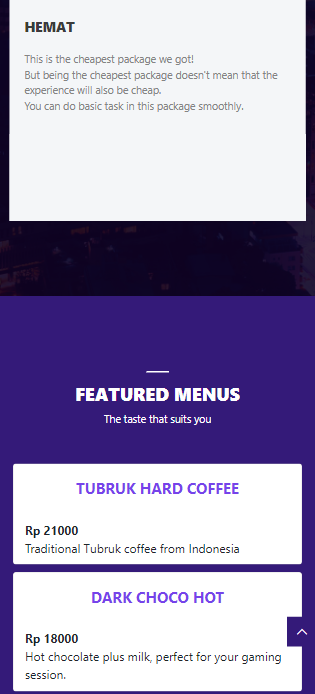


1. User register



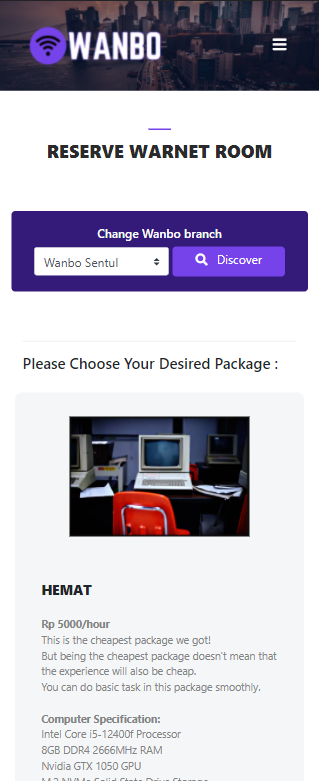
1. User login

The first design in user’s page is the register and login feature where user must fill the data required to use the Wanbo application.



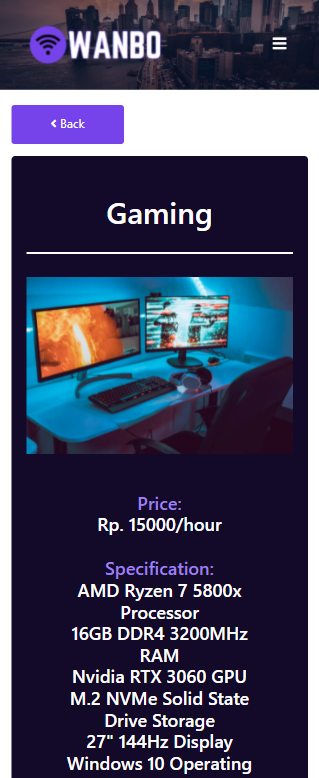
1. User dashboard

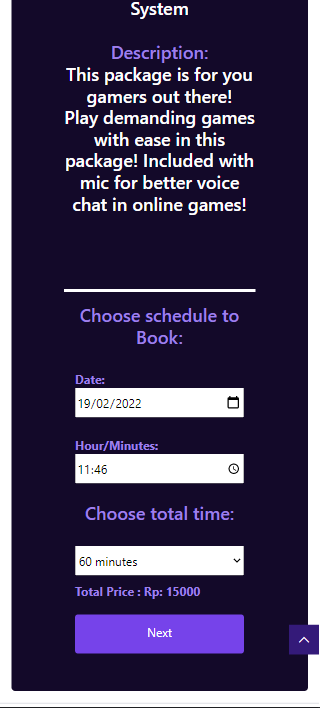
Wanbo dashboard is shown in fig.14. From the dashboard, users can see all available packages by scrolling it, and user can see all foods in the chosen store branch. Users can also go to another page like internet café detail, make a reservation, see active booking, and user profile from the dropdown menu on the top of the page.



1. Reserve page

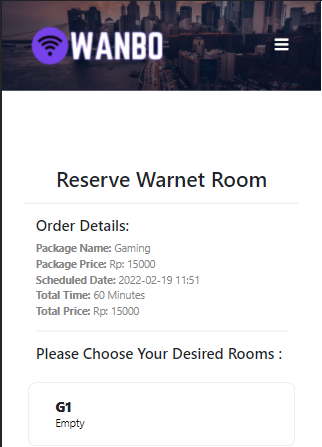
This design is the first step for users to make a booking. In this reserve menu, users can choose any package to order in the current branch or change branch to see other packages available.

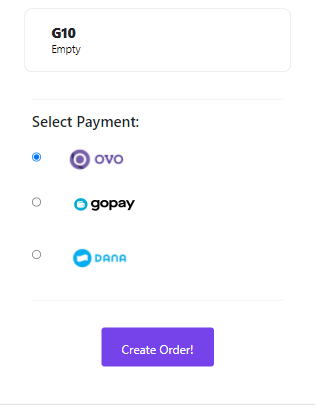




1. Detail package

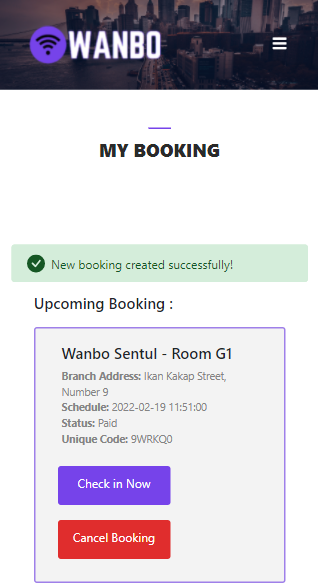
After choosing a package, users will be directed to the package's detail page. In this page, users can see package details and choose a schedule to make a booking.

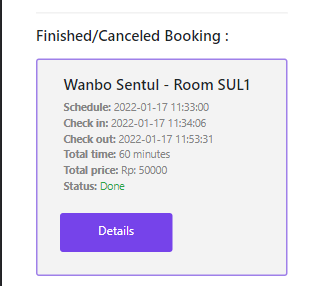




1. Confirm order

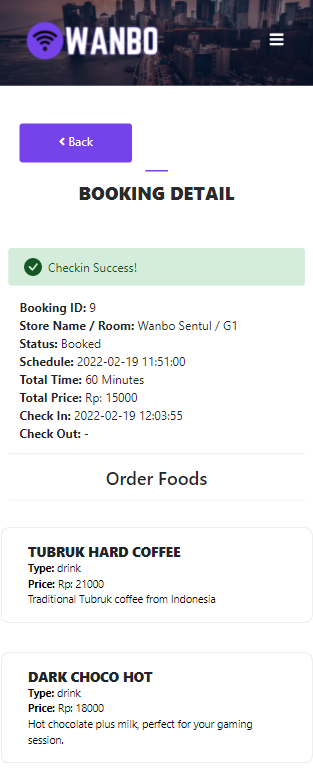
The last step in booking, users choose a room and payment type before confirming and creating the order.



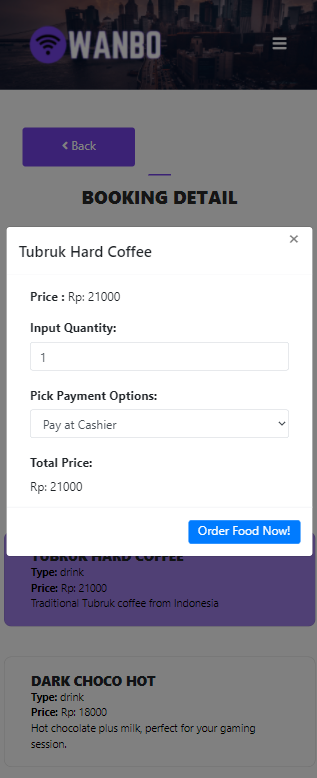


1. My booking page

After the order is created, the application is designed to direct users to the My Booking page to see, check in, or cancel the created booking and the booking history. If the booking time comes and the user is already in the Wanbo, the user can click “Check in Now”.

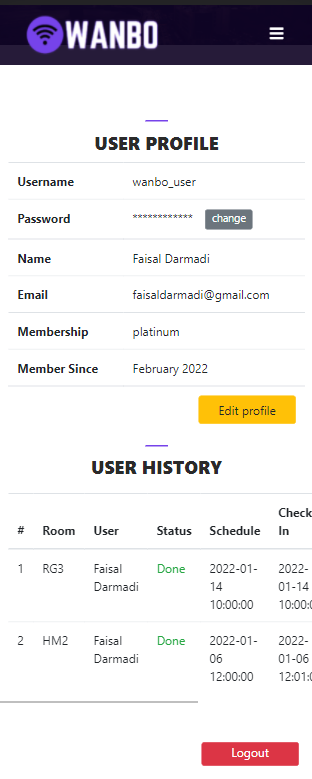


1. Ongoing booking detail



1. Order beverage(s)

While the user is using the computer, they can also order beverages from the app in the ongoing booking detail. After ordering, the food order will also display in the ongoing booking detail.



1. User profile

This is the user profile design where the users can view their profile and history. Users also can change their profile except the username because it is unique and unchangeable. The history will show the completed or cancelled booking in a scrollable table.

## Build by Feature

The last step is to build what has been designed and planned in the previous step. Before build the application started, project management is established first using Function Point (FP) metric and Constructive Cost Model (COCOMO). To calculate the COCOMO as in (1), (2), and (3), the FP must be calculated first using Relative Complexity Adjustment Factor (RCAF) and unadjusted function point (UFP) as in (4) to determine the number of KLOC.



2

Where the constant a, b, c, and d are shown in TABLE IV.

1. Constant

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Software Project** | **a** | **b** | **c** | **d** |
| Organic | 2.4 | 1.05 | 2.5 | 0.38 |
| Semi-detached | 3.0 | 1.12 | 2.5 | 0.35 |
| Embedded | 3.6 | 1.20 | 2,5 | 0,32 |

3

4

RCAF formula is shown in (5).

5

Where is the sum of 14 questions answer shown in TABLE V with number range from 0 to 5 and 0 means “no influence” and 5 means “essential”. UFP is calculated by counting the sum of external input (EI), external output (EO), external inquiry (EQ), internal file (ILF), and external interface (EIF), each multiplied by its weighting shown in TABLE IX.

1. Calculation

|  |  |  |
| --- | --- | --- |
| **No** | **Value Adjustment Factor** | **Output** |
| 1. | Does the system require reliable backup and recovery? | 3 |
| 2. | Are specialized data communications required to transfer information to or from the application? | 1 |
| 3. | Are there distributed processing functions? | 4 |
| 4. | Is performance critical? | 4 |
| 5. | Will the system run in an existing, heavily utilized operational environment? | 4 |
| 6. | Does the system require online data entry? | 5 |
| 7. | Does the online data entry require the input transaction to be built over multiple screens or operations? | 4 |
| 8. | Are the ILFs updated online? | 4 |
| 9. | Are the inputs, outputs, files, or inquiries complex? | 3 |
| 10. | Is the internal processing complex? | 3 |
| 11. | Is the code designed to be reusable? | 2 |
| 12. | Are conversion and installation included in the design? | 0 |
| 13. | Is the system designed for multiple installations in different organizations? | 0 |
| 14. | Is the application designed to facilitate change and ease of use by the user? | 5 |
|  | | 42 |

Based on (5),

1. User Page EI, EO, and, EQ Calculation

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Page** | **Type** | **Detail** |
| 1 | Login | EI | from login |
| 2 | Signup | EI | form signup |
| 3 | Guest Dashboard | EO | show features |
| 4 | Dashboard | EO | wanbo's beverages menu |
| 5 | Dashboard | EI | choose branch form |
| 6 | Dashboard Branch | EO | branch's packages |
| 7 | Dashboard Branch | EO | branch's menu |
| 8 | Dashboard Branch | EI | choose branch form |
| 9 | Profile | EO | user's data |
| 10 | Profile | EO | user's history |
| 11 | Profile | EQ | change password prompt |
| 12 | Profile | EQ | logout prompt |
| 13 | Profile | EQ | success edit password or profile prompt |
| 14 | Profile | EI | old password form |
| 15 | Profile | EI | logout form |
| 16 | Change Password | EI | change password form |
| 17 | Edit Profile | EO | user's old data |
| 18 | Edit Profile | EI | new name and email |
| 19 | Warnet | EO | wanbo's available branch(es) |
| 20 | Reserve | EO | branch's packages |
| 21 | Reserve | EI | choose branch form |
| 22 | Reserve | EQ | confimation prompt |
| 23 | Reserve Package | EO | package's details |
| 24 | Reserve Package | EI | reservation schedule form |
| 25 | Reserve Room | EO | order's detail |
| 26 | Reserve Room | EO | available rooms |
| 27 | Reserve Room | EO | available payments |
| 28 | Reserve Room | EI | select room form |
| 29 | Reserve Room | EI | select payment form |
| 30 | Reserve Room | EQ | confimation prompt |
| 31 | My Booking Ongoing Detail | EI | Order Food |
| 32 | My Booking Ongoing Detail | EI | Input Food Order Detail |
| 33 | My Booking Ongoing Detail | EO | Booking Detail |
| 34 | My Booking Ongoing Detail | EQ | Ordering Food |
| 35 | My Booking Ongoing Checkout | EQ | Confirmation check out |
| 36 | My Booking Upcoming Check In | EQ | Confirmation Check in |
| 37 | My Booking Upcoming Cancel | EQ | Confirmation cancel booking |
| 38 | My Booking Finished Booking Detail | EO | Booking Order History Detail |
| **SUMMARY** | | **EI** | **14** |
| **EO** | **15** |
| **EQ** | **9** |

1. Admin Page EI, EO, and, EQ Calculation

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Page** | **Type** | **Detail** |
| 1 | Login | EI | form login |
| 2 | Dashboard | EO | Billing info |
| 3 | Dashboard | EO | Upcoming booking |
| 4 | Dashboard | EI | input new booking |
| 5 | Dashboard | EI | Stop billing |
| 6 | Dashboard | EQ | Stop billing confirmation |
| 7 | Dashboard | EQ | Stop billing success notification |
| 8 | Billing Detail | EO | Billing Info (Detail) |
| 9 | Food order list | EO | Food Order Table |
| 10 | Food order list | EI | Accept / Decline Order |
| 11 | Profile | EO | Profile Info |
| 12 | Profile | EQ | insert password confirmation |
| 13 | Profile | EI | Update Profile |
| 14 | Beverage | EO | Beverage Table |
| 15 | Beverage | EI | Insert new beverage |
| 16 | Beverage | EO | beverage Detail |
| 17 | Beverage | EI | Update Beverage |
| 18 | Beverage | EI | Delete Beverage |
| 19 | Beverage | EQ | Delete Confirmation |
| 20 | Beverage | EQ | Success / Failed message (input / edit / delete) |
| 21 | Package | EO | Package Table |
| 22 | Package | EI | Insert new Package |
| 23 | Package | EO | Package Detail |
| 24 | Package | EI | Update Package |
| 25 | Package | EI | Delete Package |
| 26 | Package | EQ | Delete Confirmation |
| 27 | Package | EQ | Success / Failed message (input / edit / delete) |
| 28 | Room | EO | Room Table |
| 29 | Room | EI | Insert new Room |
| 30 | Room | EO | Room Detail |
| 31 | Room | EI | Update Room |
| 32 | Room | EI | Delete Room |
| 33 | Room | EQ | Delete Confirmation |
| 34 | Room | EQ | Success / Failed message (input / edit / delete) |
| 35 | Daily report | EO | - |
| 36 | Payment History | EO | - |
| 37 | Order History | EO | - |
| 38 | Food Order History | EO | - |
| 39 | Logout | EI | - |
| 40 | Logout | EQ | Logout confirmation |
| **SUMMARY** | | **EI** | **15** |
| **EO** | **15** |
| **EQ** | **10** |

1. ILF and EIF Calculation

|  |  |  |
| --- | --- | --- |
| **Measurement Parameter** | **Detail** | **Count** |
| ILF | Wanbo Database | 1 |
| EIF | API payment (ovo, dana, and gopay) | 3 |

1. UFP Calculation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Count | Simple | Average | Complex | Total |
| EI | 29 | 3 | 4 | 6 | 116 |
| EO | 30 | 4 | 5 | 7 | 150 |
| EQ | 19 | 3 | 4 | 6 | 57 |
| ILF | 1 | 7 | 10 | 15 | 15 |
| EIF | 3 | 5 | 7 | 10 | 15 |
| Total | | | | | 353 |

Based on (4),

Because mostly this application was build using PHP programming language, the LOC / FP is 67 [7]. So, the number of KLOC is estimated using (6).

6

Based on (6),

After getting the number of lines of code, the COCOMO can be calculated based on (1), (2), (3), and with consideration that Wanbo is semi-detached software project.

person-months

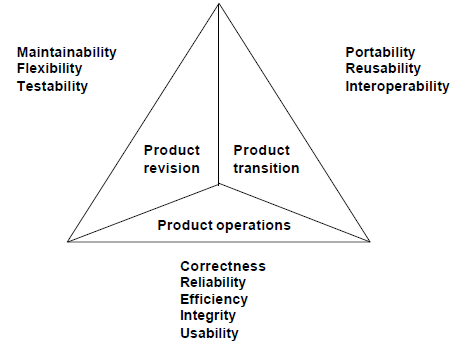
months

persons

So, from the calculation above, the Wanbo application is estimated to be built with 111.845 person-months effort, in 13 months. The estimation person needed to build this application is approximately 9 persons. But the actual time needed to complete Wanbo application is only 21 days with 3 persons.

Wanbo's backend will be built using Laravel framework because it makes the development easier using libraries and helpers they provided in this framework. MySQL will be chosen for storing all the application data, Laravel framework also provides eloquent, plus MVC model, which makes the development easier. For the frontend, we use bootstrap 5 for design and using JQuery to make the website more interactive, Font Awesome also used to implement icons in wanbo's website.

After the development process is completed, to determine the software quality, Wanbo uses McCall Triangle of Quality. This triangle also useful for the testing phase because it can make sure the software quality is good.



1. McCall Triangle of Quality
2. *Maintainability:* The effort required by users and maintenance to identify software failures, to correct failures, and to verify the success of repairs that have been made.

Wanbo application is using the Laravel framework which provides a debug feature where if there is a problem with the code, it will make the maintenance process easier and less time consuming.

1. *Flexibility:* The capability and effort required to support modification, reconfiguration, and maintenance according to user requirements.

With Wanbo, internet café admins can change the food, drink, package and room information according to their wishes and circumstances of each branch.

1. *Testability:* With regard to testing information systems with their operations to ensure specific requirements are met.

Various scenarios that may be carried out on the application every time there is a feature addition or change to ensure the application runs properly.

1. *Portability:* The tendency to adapt system software to other environments such as different hardware or different OS.

The application is made responsive so that it can be opened on a cell phone or on a PC without experiencing a drastic decrease in UI or UX.

1. *Reusability:* With regard to the transfer of modules or programs to other applications.

CRUD (Create, Read, Update, Delete) pages on Wanbo can be used for CRUD in other applications, only by changing the variables and the number of columns.

1. *Interoperability:* Focus on creating interfaces with other software systems or with other firmware tools.

Wanbo also cooperates with other software systems in our payments that can use OVO, Gopay, and Dana.

1. *Correctness:* The extent to which the program conforms to its specifications.

The program is made based on the specifications collected in the project overview and feature list, and is carried out according to planning and design by feature so that it fits the specifications.

1. *Reliability:* With regard to reliability to provide services.

In terms of reliability, Wanbo has gone through a testing process so that it is free from malfunction and continues to run as long as there are no problems with the server.

1. *Efficiency:* Efficient use of computer code to perform operations and efficient use of computer resources.

Wanbo is made using the Laravel framework which already provides many helpers that make it easier for programmers to build an application

1. *Integrity:* With regard to system security from software, preventing access from unauthorized persons.

In Wanbo, the admin page cannot be accessed by normal users and normal user accounts also cannot access other user accounts.

1. *Usability:* Can be used easily by ordinary people and can understand the software easily.

The UI is designed as simple as possible so that it is easy to understand and comfortable to use by ordinary people.

Here is the application code result:

<https://github.com/FOR3ST321/wanbo>

# Conclusion

From the development process that has been done, it can be concluded that Feature Driven Development is very helpful in building a software because it is effective, clear, and structured. It is like a framework that guides the developer to develop a software structure, especially in a short time. This methodology helps developers to prioritize and arrange the time management for each feature so that the software can be completed before the deadline.

# References

1. F. Bjorn, K. Stein and W. Fathul, “Information dissemination in a developing society: Internet café users in Indonesia”, 2005.
2. O. A. Santosa and H. Setiaji, “Pengembangan aplikasi ecommerce dengan metode feature driven development”, 2020.
3. S. A. K. Ghayyur, A. Razzaq, Z. Syed, S. Ahmed and R. Ullah, “Evaluation for feature driven development paradign in context of architecture design augmentation and prespective implication”, 2018.
4. R. Lynn, “What is FDD in agile?” [online]. Available:: <https://www.planview.com/resources/articles/fdd-agile/> [Accessed 2 November 2021].
5. Databoks Katadata (2016). [online]. Available: <https://databoks.katadata.co.id/datapublish/2016/08/08/jumlah-warnet-2014> [Accessed 2 November 2021].
6. A. K. Kanchinkoti, R. Ali, I. Walayat and Y. Bashir, “Online resaturant meal reservation system”, 2021.
7. J. Tania. (2005). [online]. Available: <https://www.cs.helsinki.fi/u/taina/ohtu/fp.html> [Accessed 27 January 2022].