

**CS251-Assignment1-part1,2**

**Date: Feb25/2025**

Cairo University

FCAI

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **ID** | **Work** | **Emails** |
| Mahmoud Sayed | 20230372 | Snake Game | [yonesmahmoud538@gmail.com](mailto:yonesmahmoud538@gmail.com)  Leader’s phone number:  01000081851 |
| Amr Khaled | 20230271 | Bank System Simulator | [amrk4912@gmail.com](mailto:amrk4912@gmail.com) |
| Saif Omar | 20230183 | Car Rental System | [saifomar2340@gmail.com](mailto:saifomar2340@gmail.com) |

[Due Date]

# Table of Contents

[CS251: Introduction to Software Engineering 1](#_Toc973213495)

[Part 1 Report: Learning Java and Project Development 1](#_Toc681303317)

[1. Chosen Language 1](#_Toc1554876470)

[2. Team Members and Learning Details 1](#_Toc351953287)

[3. Project Descriptions 1](#_Toc6966690)

[1. Banking System (By Amr Khaled, ID: 20230271) 1](#_Toc723747578)

[2. Snake Game (By Mahmod Sayed, ID: 20230372) 1](#_Toc1790736007)

[3. Car Rental System (By Saif Omar, ID: 20230183) 2](#_Toc2141851673)

[4. Screenshots & Video Links 2](#_Toc773678306)

[5. Main Function of Each Project 2](#_Toc516523469)

[Banking System (Amr Khaled) 2](#_Toc779914794)

[Snake Game (Mahmod Sayed) 2](#_Toc1821498162)

[Car Rental System (Saif Omar) 3](#_Toc1706924088)

[Evaluation of low code no code: 4](#_Toc758532469)

[What they can do: 4](#_Toc110267029)

[Application Development: 4](#_Toc831376593)

[Workflow Automation: 4](#_Toc1458612260)

[Data Management: 5](#_Toc2062066156)

[Website and Webpage Creation: 5](#_Toc541872488)

[Rapid Prototyping: 5](#_Toc364011658)

[Intgration with Other Services: 5](#_Toc960537349)

[Examples of What They Can Do: 5](#_Toc79983642)

[Will they take away all the coder jobs? 5](#_Toc939632454)

[Lets take Bubble and Bildr as a sample: 6](#_Toc724751755)

[ What they do: 6](#_Toc2077639507)

[ Benefits: 6](#_Toc1313970259)

[ System Quality: 6](#_Toc1415290806)

[ Developer Jobs: 6](#_Toc45496431)

[Bubble vs. Bildr: A Simple Comparison: 7](#_Toc1653127619)

[ Bubble: 7](#_Toc1442127313)

[ Bildr: 7](#_Toc269658839)

[ Features: 7](#_Toc478857467)

[In simpler terms: 8](#_Toc2123430630)

[Comparison: 8](#_Toc1824535309)

[Resources: 10](#_Toc830810754)

# CS251: Introduction to Software Engineering

Part 1 Report: Learning Java and Project Development

### **1. Chosen Language**

For this task, our team decided to learn and develop projects using **Java**.

### **2. Team Members and Learning Details**

|  |  |  |  |
| --- | --- | --- | --- |
| **Team Member** | **ID** | **Hours Spent Learning** | **Learning Sources** |
| Amr Khaled | 20230271 | 16 hours | Desouki, Neseem, Java Documentation |
| Mahmoud Sayed | 20230372 | 14-19 hours | W3Schools, Bro Code youtube channel, Java Documentation |
| Saif Omar | 20230183 | 1 Day | W3Schools, Desouki, Java Documentation |

### **3. Project Descriptions**

Each team member developed a unique Java application. Below is a description of each project:

#### **1. Banking System (By Amr Khaled, ID: 20230271)**

This project is a **banking management system** that allows users to register, log in, deposit, withdraw, and transfer money. It features a **transaction history system**, user authentication, and a simple menu-driven interface. The system uses file handling to save user accounts and transactions, ensuring data persistence between sessions. The core logic is built around Java **OOP principles**, particularly **encapsulation and serialization**.

#### **Snake Game (By Mahmod Sayed, ID: 20230372)**

Snake Game is a classic game loved by many people, in the game the player controls a snake by moving it on the screen to collect food. The more the snake eats, The more it grows and the more it grows, controling the snake becomes more difficult as the player is trying to prevent the snake from bumbing into the border of the frame or the body of the snake it self.

#### **Car Rental System (By Saif Omar, ID: 20230183)**

### **4. Screenshots & Video Links**

Each team member has provided **6-10 screenshots** of their project and a **3-minute video** demonstrating the functionality.

|  |  |  |
| --- | --- | --- |
| **Team Member** | **Screenshots** | **Video Link** |
| Amr Khaled | [Code ScreenShots](https://drive.google.com/drive/folders/1Uf4HtL-VxuLbFnWtd4pYlScODe9hchIy?usp=sharing) | [Bank System Simulator by Amr Khaled](https://drive.google.com/file/d/1obkdkeLPrZZxwMjQJwRUbPGfHqqfd9t9/view) |
| Mahmoud Sayed | [Code and Game ScreenShots](https://drive.google.com/drive/folders/1zDbfr3qKDuIvTw8lp-WatxX_a4enkM4y?usp=drive_link) | [Snake Game by Mahmoud Sayed](https://drive.google.com/drive/u/0/folders/19UqSgeiD0T2fmro1lJjZtkfw2drbsDR6) |
| Saif Omar | [Code and ScreenShots](https://drive.google.com/drive/folders/1I4Ie2u7TNfGnB5DL6UcLXcAm8FojdIIc) | [Car rental by Saif eldeen omar](https://drive.google.com/file/d/1jyNVeN1IX7vL7dp-DB0N3mdxGuvA2_ZA/view?usp=sharing) |

### **5. Main Function of Each Project**

#### **Banking System (Amr Khaled)**

The main function of this program initializes the banking system, loads existing user accounts from a file, and presents a menu to the user. Based on user input, it calls appropriate methods to handle registration, login, deposits, withdrawals, transfers, and transaction history.

#### **Snake Game (Mahmod Sayed)**

The main function creates a JFrame object by the name frame and sets it visible with size of $frame\_h and $frame\_w which are two values initialized in the main function with the value 600 for both, the frame is altered to open in the center of the screen by setting the location relativity to null, after that the main function create an object of type Game by the name “snake game”, the class Game extends the class JPanel and is the manager of the logic needed by the game. after creating the snake\_game the main function starts it and adds it to the frame then pack every thing in it’s preffered size using frame.pack the it uses the requestFocus method on the snake\_game instance to listen to input from the keyboard.

#### **Car Rental System (Saif Omar)**

The Car Rental System is a Java-based console application that allows users to rent cars and administrators to manage the available car inventory. It provides functionalities for selecting car types, colors, and models, calculating rental prices, and offering discounts for long-term rentals. The system also includes an admin section where new cars can be added, prices can be updated, and existing cars can be removed.

# Evaluation of low code no code:

Low-code tools use user-facing interface that enables your organization’s development team to speed up their work and reduce deployment time. No-code platforms allow citizen developers to manage and quickly deploy business applications

 no code provides drag-and-drop interfaces and pre-built components that users can combine to create simple software applications

## What they can do:

No-code and low-code tools have significantly broadened the scope of what individuals and businesses can achieve in software development

### **Application Development:**

* They enable the creation of web applications, mobile apps, and desktop applications without extensive coding knowledge.
* This includes building user interfaces, defining workflows, and managing data.

### **Workflow Automation:**

* These tools excel at automating repetitive tasks and business processes.
* Users can design automated workflows that connect different applications and services.

### **Data Management:**

* Many platforms provide tools for creating and managing databases, allowing users to store and retrieve data easily.
* They facilitate data integration, enabling data to flow between different applications.

### **Website and Webpage Creation:**

* No-code platforms simplify the process of building websites and landing pages, often with drag-and-drop interfaces and pre-built templates.

### **Rapid Prototyping:**

* They allow for quick creation of prototypes, enabling businesses to test ideas and gather feedback rapidly.

### **Intgration with Other Services:**

* Many platforms offer integrations with popular third-party services, such as CRM systems, payment gateways, and social media platforms.

### **Examples of What They Can Do:**

* Build Customer Relationship Management (CRM) Systems
* Create E-commerce Platforms
* Develop Internal Tools
* Automate Data Entry and Processing
* Build Mobile Applications for Internal or External Use

### **Will they take away all the coder jobs?**

In my opinion,No, they help users to do their tasks and jobs quickly

better and faster, Coders can use them to build the basic stuff quicker, and then focus on the really tricky and we should upgrade it

## Lets take Bubble and Bildr as a sample:

Both Bubble and Bildr are "no-code" or "low-code" platforms. This means they let you build web apps and sometimes mobile apps without writing a ton of traditional code.

### **What they do:**

* + They give you visual tools to design how your app looks and works.
  + You can connect to databases and other services.
  + You can create user logins, forms, and complex workflows.

### **Benefits:**

* + Faster development: You can build things quicker than with coding.
  + Lower cost: You don't always need a big team of developers.
  + More accessible: People with less coding skill can create apps.

### **System Quality:**

* + The quality depends a lot on how well you use the tools.
  + You can make good, working apps, but complex, high-performance systems might still need coding.
  + They are very good for MVP's (Minimum viable products) to test ideas quickly.

### **Developer Jobs:**

* + These tools won't totally replace developers.
  + They change the job: developers can use these tools to build faster, or focus on the more complex parts that still need coding.
  + They also create a need for developers that can work within the no-code enviroments, and create plugins to extend the no-code platforms.

## Bubble vs. Bildr: A Simple Comparison:

**Here's a basic look at what each platform offers:**

### **Bubble:**

* + Known for its strong database and workflow features.
  + A lot of community support and plugins.
  + Can handle complex logic.
  + Has a large user base, and has been around for a long time.

### **Bildr:**

* + Focuses on visual design and building responsive layouts.
  + Good for creating user interfaces.
  + Tries to be very flexible with its design.
  + Is newer than bubble, and is still actively developing new features.

### **Features:**

* + Both have drag-and-drop interfaces.
  + Both let you connect to APIs.
  + Bubble might be stronger for complex back-end logic.
  + Bildr might be stronger for detailed front-end design.
  + Both allow for the creation of custom workflows.
  + Both allow for database integration.

## In simpler terms:

* Think of Bubble as good for making the "brain" of your app work well.
* Think of Bildr as good for making the "face" of your app look good.

They both have their uses, and the best choice depends on what you want to build.

Sources and related content

## Comparison:

|  |  |  |
| --- | --- | --- |
| **Feature** | **Bubble** | **bildr** |
| Core Focus | Strong back-end logic, database, workflows. | Detailed front-end design, visual control. |
| User Interface | Visual drag-and-drop, can be a steeper learning curve. | Highly flexible visual builder, strong on design. |
| Database Capabilities | Robust built-in database, good for complex data. | Database integrations, emphasis on connecting to data |
| Workflow Logic | Very powerful workflow editor, can handle intricate processes. | Workflow automation, customizable triggers and actions. |
| Front-End Design | Responsive design, wide range of visual elements. | Very high control over design, pixel-perfect layouts. |

|  |  |  |
| --- | --- | --- |
| **Plugin/Integration** | Large plugin marketplace, many API connections. | API connections, focused on direct integrations. |
| **Learning Curve** | Can be significant, due to depth of features. | Varies, design freedom requires attention to detail. |
| **Community Support** | Very large and active community, many resources. | Growing community, uses platforms like Discord. |
| **Ideal Use Cases** | Complex web applications, data-heavy apps. | Visually rich web apps, internal tools, custom UIs. |
| **Coding capabilities** | Little to no coding needed within the platform itself, but can integrate custom code. | Access to custom code embeding, allowing html, CSS and javascript to be used. |
|  |  |  |

**A sample app with Bubble video :**

**<https://drive.google.com/drive/folders/1sbBTbCjCd8P541OyL6jS7fpAWDW2P2f9?usp=sharing>**

**A sample app with Bubble Screen shots :**

**<https://drive.google.com/drive/folders/1Jz9RmUNriPGPJbKD3FDmU8y-c1vn5FAr?usp=sharing>**

# Resources:

<https://www.joinsecret.com/compare/bildr-vs-bubble>

<https://zeroqode.com/no-code-tools/bildr-review/>

<https://www.avoxi.com/blog/evaluating-low-code-vs-no-code-business-users/>

<https://www.appedge.io/post/no-code-low-code-web-development-tools-comparison-bubble-webflow-flutterflow-noodl-bildr-and-wappler>

<https://www.netguru.com/blog/bubble-no-code-development>

<https://www.google.com/search?q=https://www.belighted.com/blog/no-code-review>