# Food Magic: An Augmented Reality RestaurantApplication

**Parinita Vedantam[1], Sameer Kashyap[1], Prashwitha Sanil[1], Tanzeela Fathima1, Vandana Jha[1]**
[1]Department of CSE, K S School of Engineering and Management (VTU), Bengaluru, Karnataka,

**Article Info**

***Article history:***

Received month dd, yyyy
Revised month dd, yyyy
Accepted month dd, yyyy

***Keywords:***

3D,
Augmented Reality,
Virtual Reality,
ARCore.

**ABSTRACT**

Augmented Reality is an exciting technology that has been booming in recent years and might just be the solution to bridging the gap between the experiences in online and offline shopping, education, entertainment, and many more. AR can be used to market a variety of products, one of them is food. AR has been used by many big tech companies like Snapchat and Instagram to enhance the user experience using computational power. One feasible application of this technology in the advancement of the food industry is by displaying a food sample all the more alluring and how the food would look like in cafés and restaurants using 3D models of these foods. In this paper, we explore, and discuss existing systems in the food industry and challenges in developing such systems. We propose one of the feasible uses of this technology in the development of the food industry is to use 3D food models to show a food sample and how the food will appear in cafés and restaurants. Food Magic will encourage customers to view abstract food in the restaurant as more appetizing when exploring, buying, taking out or eating. We also present other important aspects, including key requirements for successful deployment of Food Magic. Finally, open research issues are outlined.

***Corresponding Author:***

Vandana Jha

[1]Department of CSE, K S School of Engineering and Management (VTU), Bengaluru, Karnataka
Email: vandanajha@kssem.edu.in

## 1. INTRODUCTION

Augmented Reality is an exciting technology that has been booming in recent years and might just be the solution to bridging the gap between the experience in online and offline shopping, education, entertainment, and many more. AR can be used to market a variety of products, one of them is food. One feasible application of this technology in the advancement of the food industry is by displaying a food sample and how the food would look like in cafés and restaurants using 3D food models. Food Magic will enable clients to perceive abstract food as more appetizing while they consider ordering, takeout or eating in the restaurant.

Augmented Reality and Virtual Reality have emerged as the most promising technology advancements in recent years. They both benefit from the recent growth in compute resources available in limited form factors. AR/VR utilizes the hardware capabilities of these machines and gives the users an immersive experience for multiple applications like gaming, media consumption, flight simulation, combat simulation, etc. Virtual Reality is a technique by which the user is placed inside a virtual world, typically by a gadget that is directly placed in front of the eyes worn over the head, called the “VR headset” giving the user an experience of actuallybeing present in the virtual world as it engages sight and hearing senses, giving the user an immersive experience. VR has been constantly evolving over the last few years and has led to major big tech companies adopting their products especially in the field of gaming and simulation.

AR is classified into two categories. A marker-based AR system searches the world for a certain image pattern and superimposes the simulated object on top of it. As a result, the AR device's camera will continuously scan the input and perform marker — image pattern recognition before creating geometry and placing virtual objects. The geometry and location of the marker will control the geometry of the virtual frame. If the camera loses sight of the marker, the simulated object is misplaced, and when it returns to view, the object is replaced. In markerless AR, the virtual object is positioned in the geometry created by SLAM (Simultaneous Localization and Mapping), which takes in the camera feed and generates a threedimensional mesh of the world. As a result, the app saves the world as a 3D model. As a result, when a virtual object is placed in an environment, it is placed in its 3D model. As a result, even though the camera loses sight when it returns, the simulated object will still be found in the same place.

The following is the format of the paper: Section II describes the related work. Section III presents our suggested system and methodology, followed by Section IV's comprehensive implementation. Section V presents the results and a thorough experimental evaluation. Section VI concludes with a proposal for future work.

## 2. RELATED WORK

This section shows the analysis and research made in the field of AR. Google business view/street view used in [1], to build a Visual Commerce Engine system using which the stores can rebuild their trust and relationship with customers offline as well as online. The proposed system. The proposed system marks/tags the product on the Google business photo/street view iFrame and places them on sale/buy. It precisely simulates the physical store's expected atmosphere, landscape, and consumer shopping experience, which uniquely differentiates the store.

TeachAR, a marker-based AR tool for teaching colors, shapes and spatial relationships to children aged 4 to 6 years [2], uses the AToolkit plugin for Unity for square marker tracking. The Mickrosoft Kinect’s microphone and speech API is used for isolated word speech recognition, a webcam for image capturing and a desktop monitor for viewing the AR scene.

AR may be used to sell a range of items, one of which is food. A system proposed in [3] is a markerless AR for food advertisements using Vuforia that contains six markers and 3D items gathered from Turbosquid and read from a food menu available for the marker to be scanned. The challenge faced by the users was not to be able to find the marker.

Markerless AR based systems increase customer usability as they improve consumer accessibility because there is no external dependency, as opposed to Marker-Based AR, which requires placing markers where customers want the product to be rendered. The system proposed in [4], is implemented by using Unity Vuforia SDK which can be deployed in Android, iOS and even WebGL. The initial screen of the developed application is a list of furniture choices. After determining the ground plane, the model is made into real-world space in the camera's view plane. The optical inertial odometry of the phone sensors maintains the object's orientation[5],[6],[7].

## 3. SYSTEM DESIGN

In order to view food in AR, a representation of food items must be rendered in the user's environment and the rendered model must be tracked in the field of view. The application's parameters are summarised using a flowchart in Figure:1. The flowchart of the application in Figure 2, depicts the different phases in the workflow or procedure and how they are streamlined.

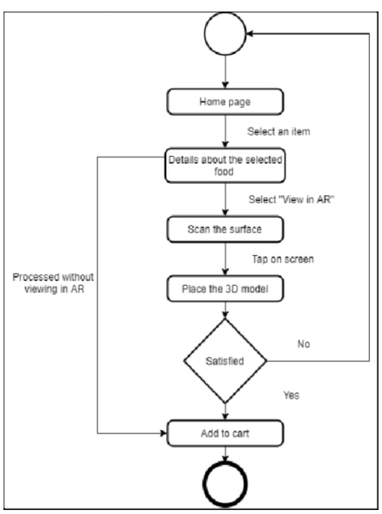


Figure 1: Flowchart for AR

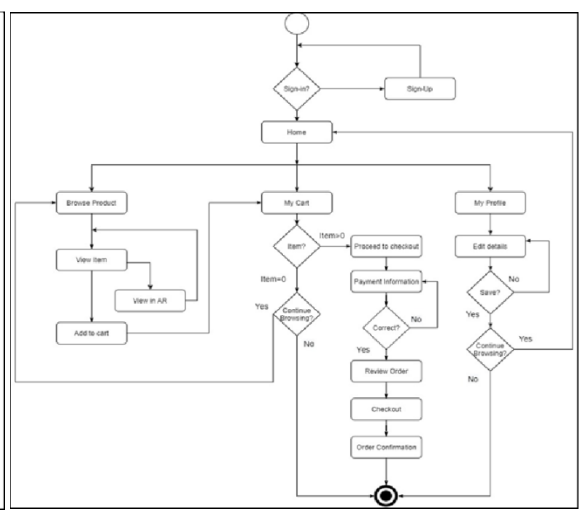


Figure 2: Flowchart of Food Magic Application

A sequence diagram is a type of interaction diagram that shows how a set of objects interact and in which order. As shown in Figure 3 is the sequence diagram of Food Magic. The rectangular boxes represent the time needed for an object to complete a task. The user first starts the app and selects a food item that the user wants to view in 3D. After selecting the "View in 3D" option, the ARCore plugin starts detecting the plane or scans the environment through the user's mobile/device camera. After detecting the plane, the fooditem is Augmented and is displayed on the users screen placing the item where the plane had been detected. Following this the user has the option to add the food item to the cart or go back to the screen before and continue browsing other food items.

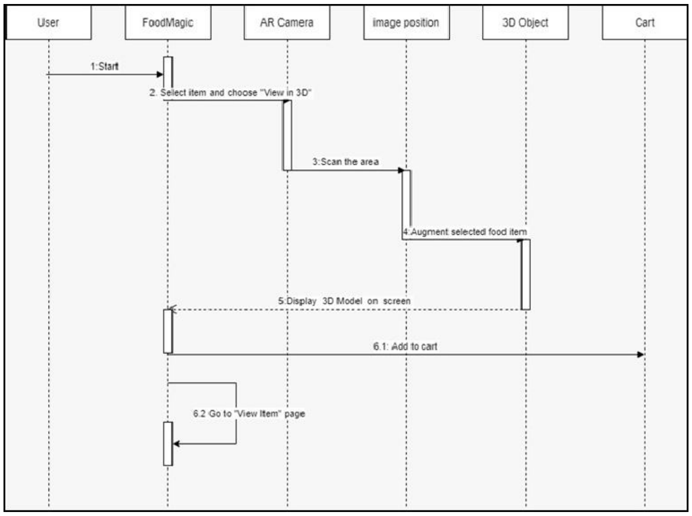


Figure 3: Sequence Diagram

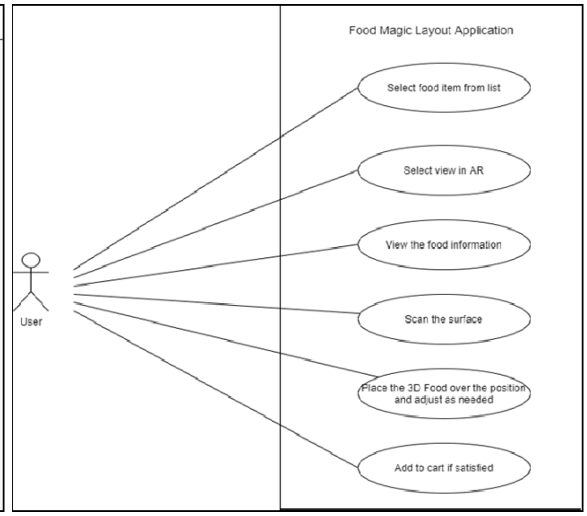


Figure 4: Use Case Diagram

A use case diagram is a visual description of a user's interaction with a device that depicts the user's relationship with the various use cases in which the user is involved. The use case diagram of an user who signs into the Food Magic App is as seen in Figure 4. It depicts the user's relationship with the device. The user chooses a food item and loads the camera to see an augmented model of the same. He then scans the surface, and after the surface has been identified, the 3D food object is positioned. The use case diagram has all of the interactions and relationships defined.

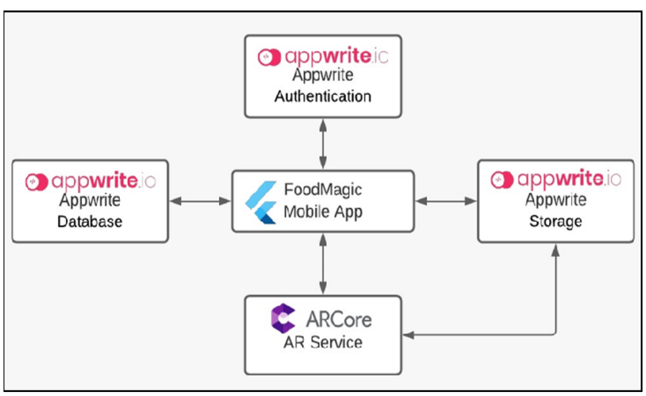


Figure 5: Software Architecture

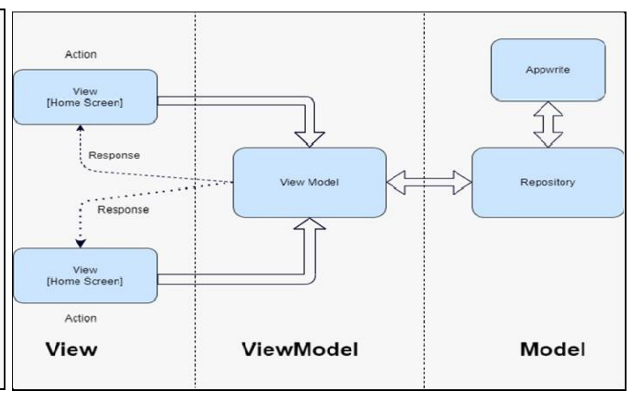


Figure 6: MVVM

Figure 5 depicts the Software Architecture, which refers to the overall system's total structure/architecture and converts software characteristics like scalability, reusability, extensibility, modularity, and maintainability into structured solutions to meet market requirements. Food Magic's software architecture consists of five main components:

1. Mobile App/Front-end:
   The proposed application, Food Magic's, frontend is created using Flutter. We chose Flutter for its fast growth, expressive and flexible user interface, and native performance that takes into account all platform differences. The Flutter front-end is built using the Model-View–ViewModel (MVVM) pattern. MVVM as shown in Figure 6, is a design pattern that separates

business logic from ViewModel and keeps only UI elements in View. The ViewModel connects the View and the Model by bringing user events and returning the result.

2. Backend - Appwrite:
The backend of Food Magic uses Appwrite. Appwrite is a new open-source, end-to-end back-end server for front-end web developers that helps you to create applications much faster. Its aim is to help developers create advanced apps faster by abstracting and simplifying basic programming activities behind REST APIs and tools. Appwrite server is bundled as a Docker container that can be installed using a single terminal command on the local computer or on a cloud provider.

3. Appwrite Authentication:
The Appwrite Authentication service simplifies device registration and login. Auth has built-in connectivity with a variety of OAuth providers, including Facebook, Github, LinkedIn, and others. The Auth service reduces the time and effort required to create a stable and safe user authentication and authorization system that can be used with a variety of third-party login methods.

4. Appwrite Database:
The Appwrite Database service allows us to create structured collections of documents, query and filter lists of documents, and manage an advanced set of read and write access permissions. All the data in the database service is stored in structured JSON documents.

5. Appwrite Storage:
We can handle all project files with the Storage service. We can upload, open, download, and query any of the project files using the Storage facility. Read and write permissions are assigned to each file in the service to control who has rights to view or update it. We can create preview images of the files using the preview endpoint. We may also use the preview endpoint to manipulate the resulting image so that it matches well inside the software in terms of dimensions, file size, and shape. The view endpoint is often used to change the document file format for better compression or image size for better network delivery.

## 4. IMPLEMENTATION OF THE SYSTEM

This section of the paper contains the execution of the theoretical design into a working system. The deployment stage is critical for maintaining a new system's efficiency and giving the customer confidence that it can work and succeed. The project has been divided into 4 phases, outlined as follows:

1. Front-end: Flutter Phase:
This stage entails using Flutter to build a front-end UI for the Food Magic application. Flutter is popular due to its fast growth, expressive and flexible user interface, and native performance that takes into account all platform differences. Flutter consists of two important parts :

    - An SDK (Software development kit) - A collection of tools that are going to help you develop your applications. This includes tools to compile your code into native machine code.
    - A Framework (UI library based on widgets) - A collection of reusable UI elements (buttons, text inputs, slides and so on) that you can personalize for your own need.Flutter uses a variety of widgets to deliver a fully functioning application. These widgets are Flutter's framework architecture.

    A. Flutter Widgets
    Flutter widgets are built using a modern framework that takes inspiration from React. The central idea is that you build your UI out of widgets

    - Widgets describe what their view should look like given their current configuration and state.
    - When a widget's state changes, the widget rebuilds its description, which the framework differs against the previous description in order to determine the minimal changes needed in the underlying render tree to transition from one state to the next.
    - A widget's main task is to implement a build() function, which describes the widget in terms of other, lower-level widgets.
    - The framework builds those widgets in turn until the process bottoms out in widgets that represent the underlying RenderObject, which computes and describes the geometry of the widget.

    During the development of the Food Magic app, widgets that are subclasses of either Stateless Widget or Stateful Widget, depending on whether the widget manages any state are used.

- Stateful Widget - It is a widget that has a mutable state. State is the information that (1), can be read synchronously when the widget is built and (2), might change during the lifetime of the widget. Stateful widgets are useful when the part of the user interface being described can change dynamically, e.g. due to having an internal clock-driven state, or depending on some system state.
- Stateless Widget - A widget that does not require mutable state. Stateless widgets are useful when the part of the user interface that is being described does not depend on anything other than the configuration information in the object itself and the BuildContext in which the widget is inflated.

B. Code Snippets

Food Magic consists of multiple pages in the application, a few of the pages are explained and outlined as follows:

1. Cart - A few important widgets used here are:

   build() : The framework calls this method when this widget is inserted into the tree in a given BuildContext and when the dependencies of this widget change. The framework replaces the subtree below this widget with the widget returned by this method, either by updating the existing subtree or by removing the subtree and inflating a new subtree, depending on whether the widget returned by this method can update the root of the existing subtree.

   Scaffold() : Implements the basic material design visual layout structure. The Scaffold widget takes a number of different widgets as named arguments, each of which are placed in the Scaffold layout in the appropriate place. appBar, body are the properties in the scaffold.

   AppBar() : A material design app bar. The AppBar displays the toolbar widgets, leading, title, and actions.

```
class CartView extends StatelessWidget {
   @override
   Widget build(BuildContext context) {
     return Scaffold(
       appBar: AppBar(
         elevation: 0,
         centerTitle: true,
         toolbarHeight: 80,
         title: Text(
           "CART ITEM",
           style: TextStyle(
            fontSize: 16, fontWeight: FontWeight.bold, color: Colors.white
    ),
        ),
         bottom: PreferredSize(
           preferredSize: const Size.fromHeight(0.0),
           child: const SizedBox(),
        ),
       actions: [
       Icon(Icons.shopping_cart).padR(15.0),
       ],
     ),
       body: CustomBackground(
         child: SingleChildScrollView(
            child: CartContent(),
        ),
      ),
    );
  }
}
```

2. Home - A few important widgets used here are:

   Container() : A convenience widget that combines common painting, positioning, and sizing widgets. The Container widget lets you create a rectangular visual element. Margin, Child are a few properties that can be configured in the container. (1)margin- Empty space to surround the decoration and child. (2)child- The child contained by the container.

   Row() : A widget that displays its children in a horizontal array. (1)children- The widgets below this widget in the tree.

TabBar() : You can create tabs using the TabBar widget.

```
Container(
   margin: const EdgeInsets.symmetric(horizontal: 15),
   child: Row(
    children: [
     TabBar(
       isScrollable: true,
       labelColor: Colors.white,
       indicatorColor: Colors.transparent,
       tabs: <Widget>[
         const Tab(text: "Pizza"),
         const Tab(text: "Burger"),
         const Tab(text: "Tacos"),
    ],
  ),
```

2. 3D Model Development: Blender Phase:
Blender is used to create 3D models of food items. Blender is an open-source system that contains one of the most robust 3D-graphics programming suites on the market. Modeling, rigging, animation, synthesis, rendering, compositing, and motion control, as well as video editing and 2D animation pipelines, are all supported. Blender 2.92 onwards has been used as it has new features like Geometry Nodes, Poisson disk algorithm, etc. to make the development process easier.

   A. Blender Interface
   Blender Interface is made up of many varieties of editors for displaying and modifying various facets of data. The number and types of windows that appear on the screen are not fixed and can be changed manually or by choosing a pre-set from the Screen Layout menu at the top of the screen. The following are some of the editors that were used to build the models:

   - 3D Viewport - The viewport is a window that helps you to see around the scene that includes your model. It is a part of the Blender interface that is just there to help you create, edit, display, and animate your model. It does not play a significant role in rendering. The 3D viewport has elements as shown in Figure 7, a grey cube (mesh), object origin, light, camera, 3D cursor, grid floor.

   - Shader Editor - Materials created with colour, pattern, and texture are referred to as materials in Blender. The Shader Editor is used to make changes to the materials used in rendering. A node tree is used to describe materials that Cycles and Eevee use. As a result, the Shader editor's main window serves as a node editor.

   - Properties - Many active data, including the active scene and object, are displayed and editable in the properties window. The Properties have several categories, which can be chosen via tabs. Each tab regroups properties and settings of a data type.

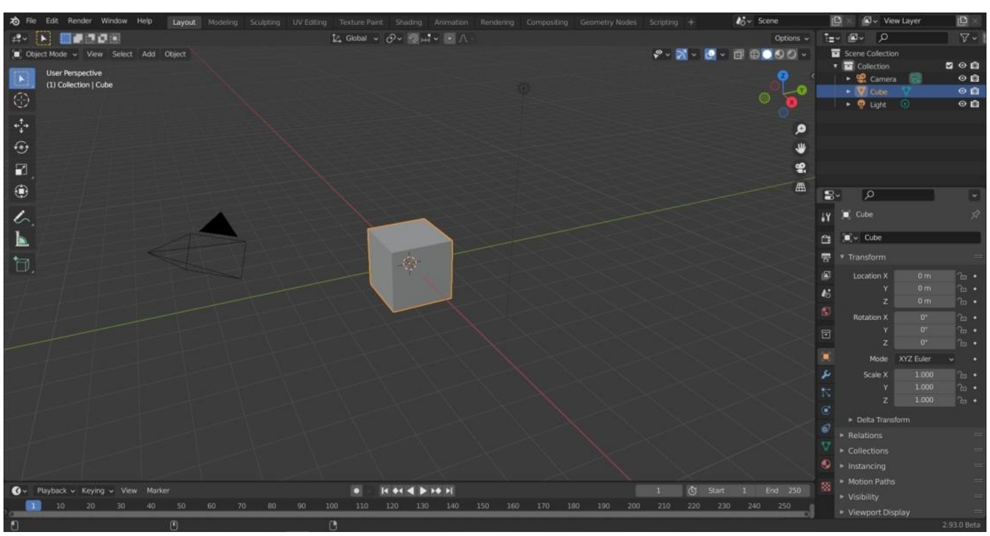

Figure 7: Default Blender Interface

B. Modeling

Modeling is the art and science of constructing a surface that either replicates the form of a real-world entity or reflects your creativity by abstract objects. It is the first stage of the implementation phase of creation of a model. In this stage we create models of food items which are used in the app. Blender's robust modelling tools make it simple to create, turn, sculpt, and edit the models. This is carried out using Meshes, Modifiers and Geometry Nodes.

- Mesh - Meshes are primitive shapes, these are basic 3D shapes like cylinder, cube, sphere, 2D shapes like circle, plane and so on. These basic shapes are modeled and edited to create a more complex object looking similar to the food items.
- Modifiers - Modifiers are non-destructive operations that are performed automatically on an entity. Modifiers allow one to automate certain results that would otherwise be too time-consuming to update manually. They work by changing how an object is displayed and rendered, but not the geometry which can be edited directly. Modifiers used most frequently in the 3D modeling for this application are:
  1. Solidify - The Solidify modifier applies depth and thickness to the surface of any mesh.
  2. Subdivision Surface - The Subdivision Surface modifier divides a mesh's faces into smaller ones, giving it a smoother look. It allows you to model plain, low-vertex meshes while creating complex smooth surfaces. It gives the object a sleek, "organic" appearance.
  3. Smooth - The Smooth modifier, smooths a mesh by flattening the angles between neighbouring faces. It smooths the mesh without subdividing it; the number of vertices remain unchanged.
  4. Cloth - The Cloth modifier is a container for a Cloth Physics simulation. It can be useful to simulate on a low-poly mesh to improve the visual quality of the cloth without drastically increasing simulation times.
  5. Collision - Collision Physics is a container for the Collision modifier. Collision physics allows various physics models to interact.

  For example, as shown in Figure 9, to obtain the cheese-like character, physics modifiers like Cloth and Collision are used. The cheese plane is given the Cloth Modifier and the patty on the bottom of the cheese is given the Collision Modifier. On applying these and starting the animation, the cloth-like cheese collides with the patty and gives the natural cheese-like effect.

- Geometry Nodes - Geometry Nodes can be used to change an object's geometry in a more complex way than regular modifiers. All vector coordinates are in world space. For volume shaders, only the position and incoming vector are available. A geometry node has no inputs or properties attached to it.


Figure 8: Modeling of Burger Model

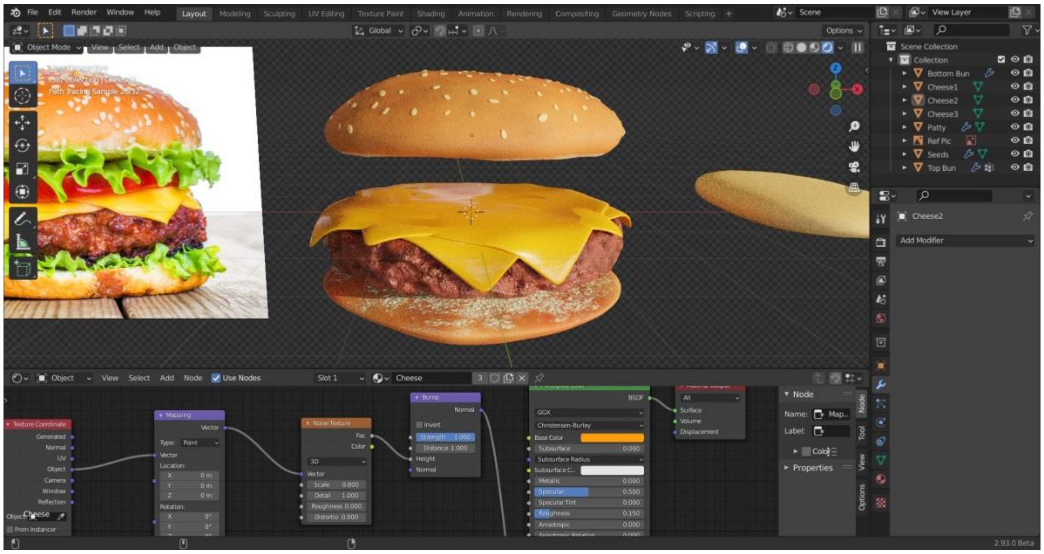


Figure 9: Cheese using Cloth Modifier

C. Materials and Shading

Materials control the appearance of meshes, curves, volumes, and other objects. They define the substance that the object is made of, its color and texture, and how light interacts with it. Materials can be created in either the Material properties, or in the Shader Editor. These provide a different view of the same shader nodes and material settings. The material system is built with physically based rendering in mind, separating how a material looks and which rendering algorithm is used to render it. This makes it easier to achieve realistic results and balanced lighting, though there are a few things to keep in mind.

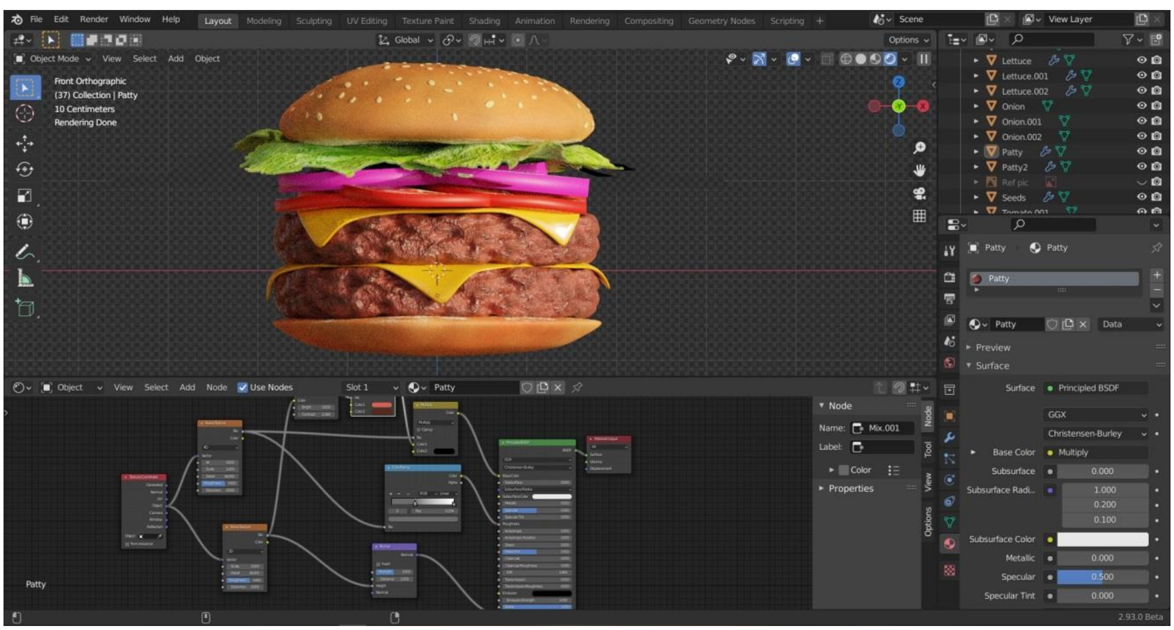


Figure 10: Materialized Burger Model

D. Rendering

After the development of the 3D models, the next step is their rendering. There are mainly three built-in Rendering Engines:

- Workbench Engine - Workbench is Blender's original engine, designed for the quickest, most graphically simple rendering during the modelling and test animation processes. It is not meant to be a render engine capable of producing final images for a project. Its main function is to display a scene in the 3D Viewport as it is being worked on. It does not calculate any lighting. The base color of the scene will be rendered.

- Cycles Engine - Cycles is a physically-based, unbiased path tracing rendering solution for Blender. The key feature of Cycles is its ability to trace paths. Path tracing, like ray tracing, is an algorithmic technique for rendering imagery by simulating how light reflects on an object. Renderman, Pixar's native rendering engine, functions similarly to Blender's Cycles engine, but where Renderman falls short is the lack of the user-friendly gui that Cycles is popular for. Cycles is the best choice for realistic rendering.

- Eevee Engine - Blender's most recent built-in rendering engine, Eevee (short for Extra Easy Virtual Environment Engine), is motivated by code close to Epic Games' Unreal Engine. Blender's most recent built-in rendering engine, Eevee (short for Extra Easy Virtual Environment Engine), is motivated by code close to Epic Games' Unreal Engine. Although it lacks Cycles' visual capability, it makes up for it with unrivalled speed.

In this project, we are using Cycles Rendering as we desire realistic looking models. Cycles' default settings are essentially photo-realistic right out of the box - they are not flawless, but they're similar enough for the majority of human eyes.

3. Integration: ARCore Phase:

ARCore is Google's tool for creating Augmented Reality applications that use extraordinary APIs to add capabilities to millions of mobile devices. It allows the device to detect the environment, identify the arena, and interact with it. There are three ways to link digital matter to the current world, as seen through the camera of your device:

1. Motion tracking: This feature allows the phone to recognise and monitor its position in relation to the rest of the world.
2. Surrounding Comprehension: This allows the phone to recognize the scale and location of all types of surfaces, including horizontal, vertical, and measured surfaces such as the ground, a footstool, or walls.
3. Light Estimation: This feature helps the unit to assess the current lighting conditions in the setting.

## 5. RESULTS AND ANALYSIS

This section deals with the results obtained after the execution of the application. The application is deployed on an Android phone here and the results are shown below. Figure 11 shows the Sign In screen of the app. Here, the user is asked to Sign in using the email and password. However, if the user does not have an account, he has the option to sign up and create an account.

The app's Home screen is displayed in Figure 12. All food products are shown here. The categories such as pizza, burger and desserts may be navigated by the user. The dish is presented with its price. The Item Detail View app screen is shown in Figure 13. When the user wishes to view the details of a particular food item and taps on it, he is redirected here. For a better understanding a brief description of the culinary item is offered. The 3D View screen of the application is displayed in Figure 14.

If the user wants to see the meal in 3D, he is taken here by clicking the 3D icon. As an example, a 3D model of farmhouse pizza is displayed on this screen in figure 14, and a 3D model of chicken burger is shown in figure 15. The Cart screen of the app is shown in Figure 16. Here are shown all the food items that the user adds to the cart with the price of the items and the total.

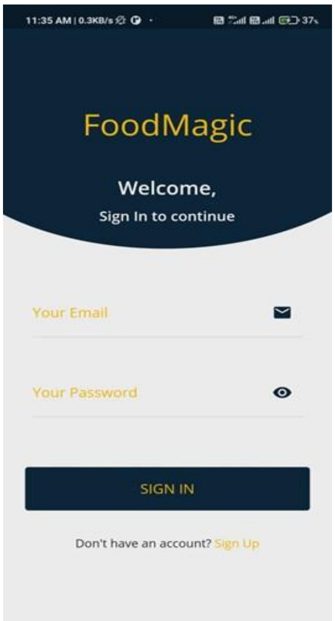


Figure 11: Sign In Screen

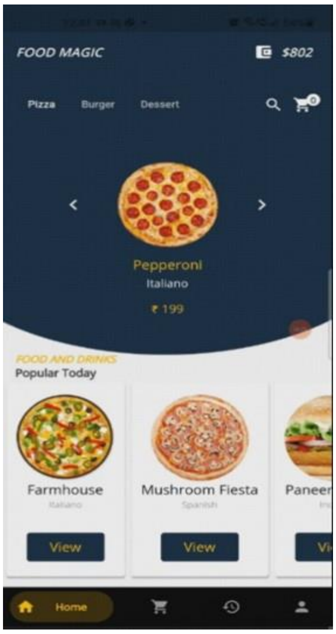


Figure 12: Home Screen


Figure 13: Item Detail View Screen

On hitting the ‘Order Now’ button, the user continues further to Orders Screen as shown in figure 17. The status of the delivery of food items requested by the user is displayed. It shows when the order is accepted, and when it is on the way. It is also shown in the app on the screen later when the order is delivered.



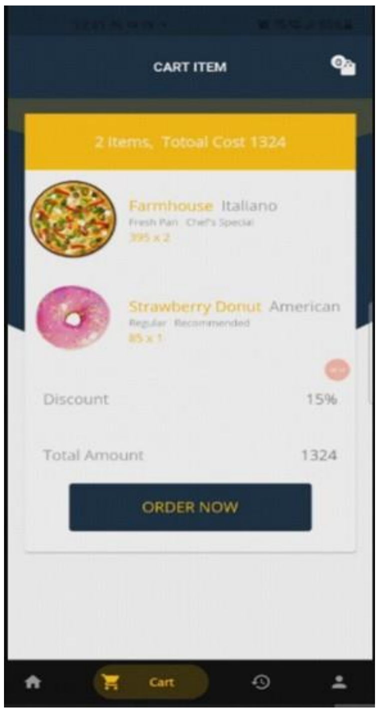

Figure 14: 3D View of Pizza Figure 15: 3D View of Burger Figure 16: Cart Screen

Figure 18 shows the app's screen for your orders. It displays all the user's recent orders. The date and time, the list and the total price of all food items are shown. The profile screen of the user is shown in Figure 19. All the personal information of the user, such as name, email id, phone number, is displayed here. The user can also write a bio and edit the details


Figure 17: Orders Screen


Figure 18: Your Orders Screen

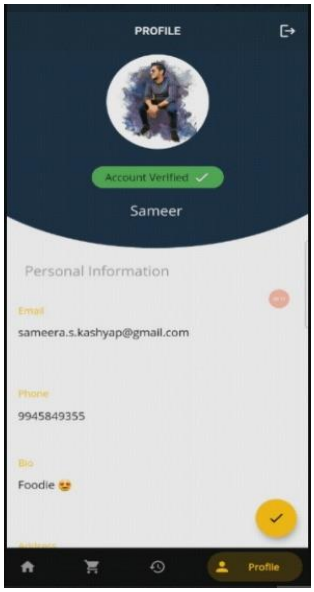


Figure 19: Profile Screen

## I. CONCLUSION AND FUTURE WORKS

The results from this project contribute to a variety of learning. Our proposed system 'Food Magic' uses AR to show food items in 3D in cafés and restaurants which encourages customers to view abstract food in the restaurant as more appetizing when exploring, buying, taking out or eating. We also present other important aspects, including key requirements for successful deployment of Food Magic in a detailed four phased implementation as well as execution. Our four phase project enabled us to work on each segment of our project separately while testing each simultaneously for best efficiency and integrating all the components as a complete working app. Finally, the tech stack used for the project [Flutter + ARCore + Blender + AppWrite] is a good fit for making efficient applications for AR.

Additional screens for the precise ingredients of each food item can be added in the future. Additional features include delivery tracking through maps, as well as a review and rating service. Model rendering quality can also be enhanced in future work by updating the implementation package.

## References

[1] M A Kodandarama, Sharath Chandrashekar, "Augmented Reality Solution for Retail Using Visual Commerce Engine", *2016 International Conference on Cloud Computing in Emerging Markets, IEEE*

[2] Che Samihah Che Dalim, Thammathip Piumsomboon, Arindam Dey, Mark Billinghurst, Shahrizal Sunar, "TeachAR: An Interactive Augmented Reality Tool for Teaching Basic English to Non-native Children", *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct), IEEE*

[3] Alvin Candra Wijaya, M. Wiranto Aris Munandar, Fitri Utaminingrum, "Usability Testing of Augmented Reality for Food Advertisement Based on Mobile PhoneUsing System Usability Scale", *2019 International Conference on Sustainable Information Engineering and Technology (SIET), IEEE*

[4] Sashank Sridhar, Sowmya Sanagavarapu, " Instant Tracking-Based Approach for Interior Décor Planning with Markerless AR", 2020 Zooming Innovation in Consumer Technologies Conference (ZINC), IEEE

[5] V. Jha, P. D. Shenoy and V. K. R, "Development of Facial Expression Classifier using Neural Networks," 2019 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE), 2019, pp. 1-4, doi: 10.1109/WIECON-ECE48653.2019.9019937.

[6] S. A. Kulkarni, V. D. Raikar, B. K. Rahul, L. V. Rakshitha, K. Sharanya and V. Jha, "Intelligent Water Level Monitoring System Using IoT," 2020 IEEE International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC), 2020, pp. 1-5, doi: 10.1109/iSSSC50941.2020.9358827.

[7] Sandhya.A. Kulkarni, C S Sowmya, P Subhalakshmi, S A Tejashwini, V R Sanusha, S Amithaand Vandana Jha, "Design and Development of Smart Helmet to Avoid Road Hazards Using IoT," *2020 IEEE International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC)*, 2020, pp. 1-6, doi: 10.1109/iSSSC50941.2020.9358838.

## Authors

**First Author** – Parinita Vedantam, CSE, K S School of Engineering and Management (VTU), vedantampari@gmail.com.

**Second Author** – Sameer Kashyap, CSE,K S School of Engineering and Management (VTU), sameera.s.kashyap@gmail.com.

**Third Author** – Prashwitha Sanil, CSE, K S School of Engineering and Management (VTU), prashsanil03@gmail.com.

**Fourth Author** – Tanzeela Fathima, CSE, K S School of Engineering and Management (VTU), 10zeela786fathima@gmail.com.

**Fifth Author** – Vandana Jha, CSE, K S School of Engineering and Management (VTU), vandanajha@kssem.edu.in