CSE / EEE / ETE 499A (Section 02)

Design Report (CO2)

Project Title: ***TollKeeper.ai***, an automated toll payment system

Submitted To

Dr. Shazzad Hosain (SZZ)

Date: 17/09/2023

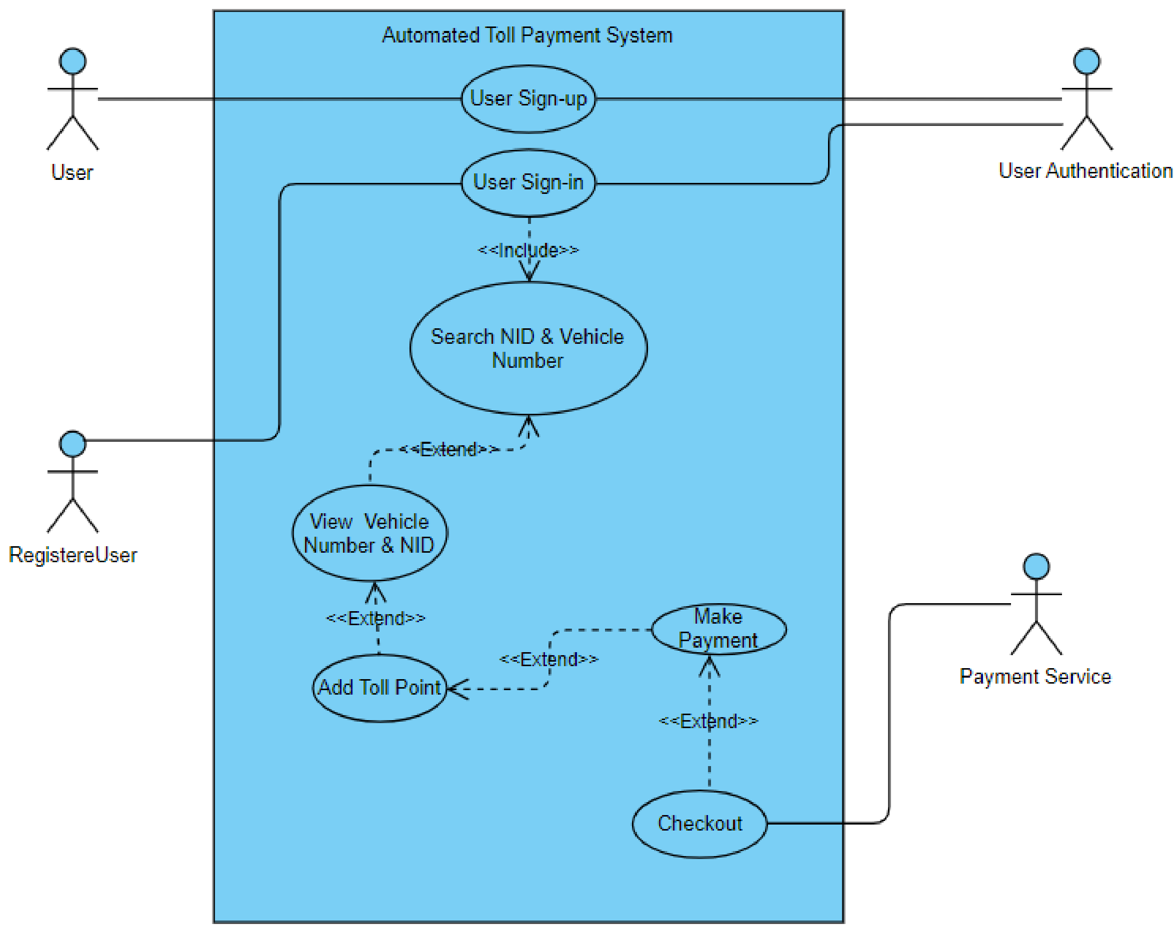
Group No: G-8

Group Members

ID	Name
2012136042	Farzan Noor Chowdhury
2011719642	Sanjena Akhter
2013680642	Jannatul Ferdous

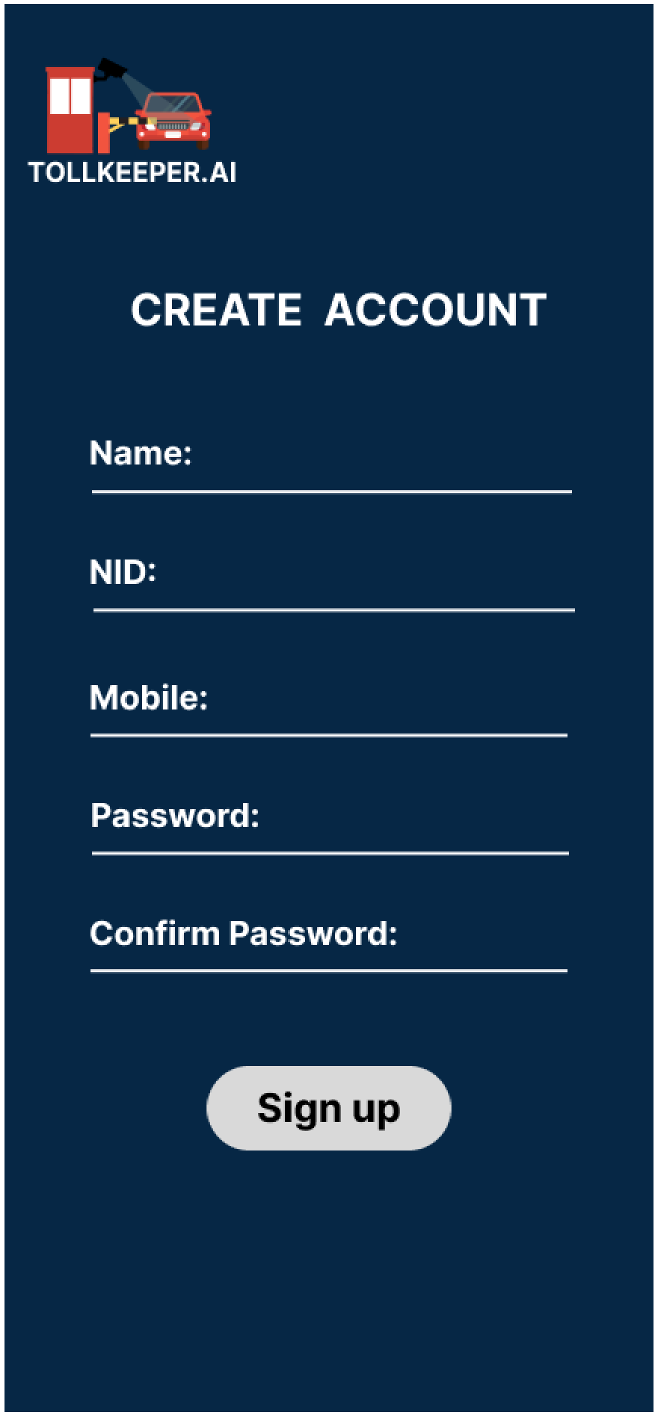
CO2: Design a system, component or process to meet desired needs within realistic constraints

1. Use Case Diagram

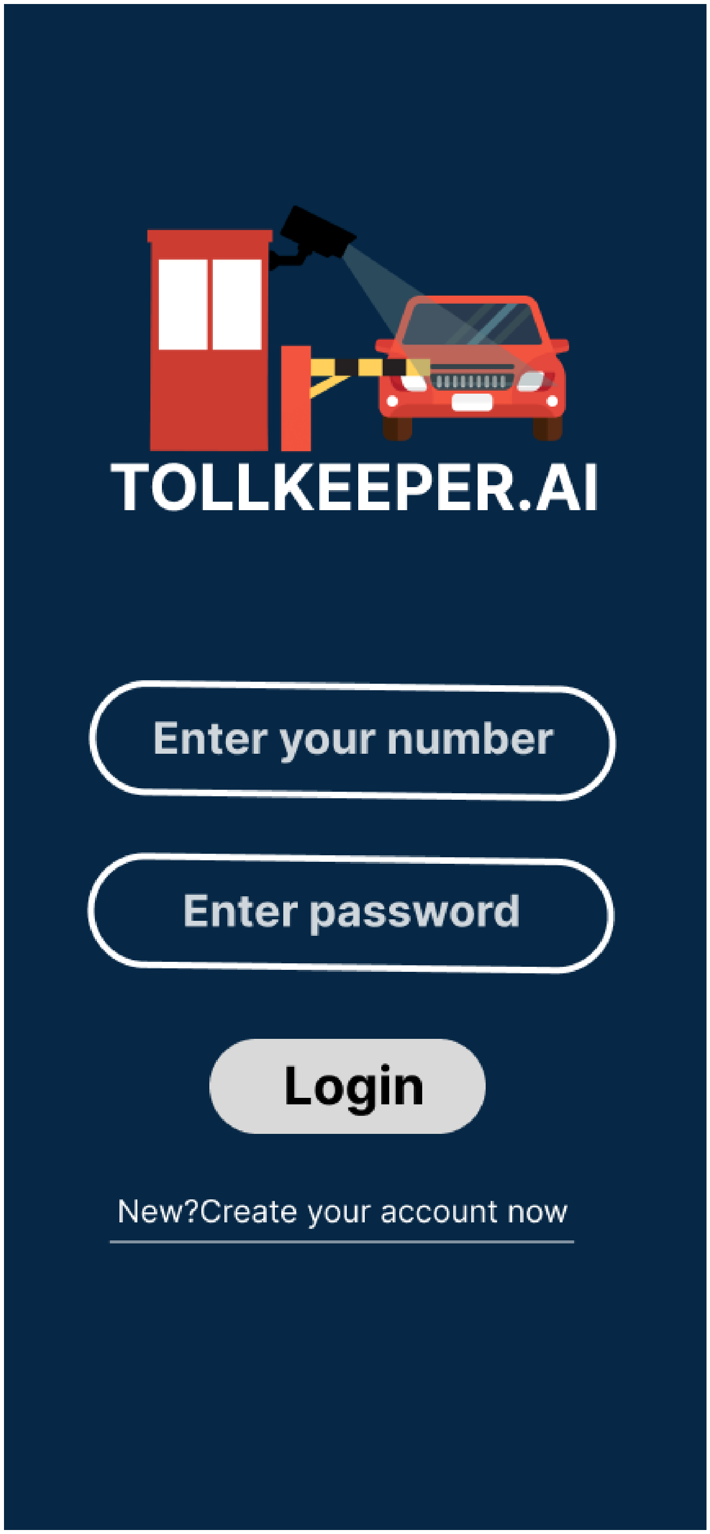


CO2: Design a system, component or process to meet desired needs within realistic constraints

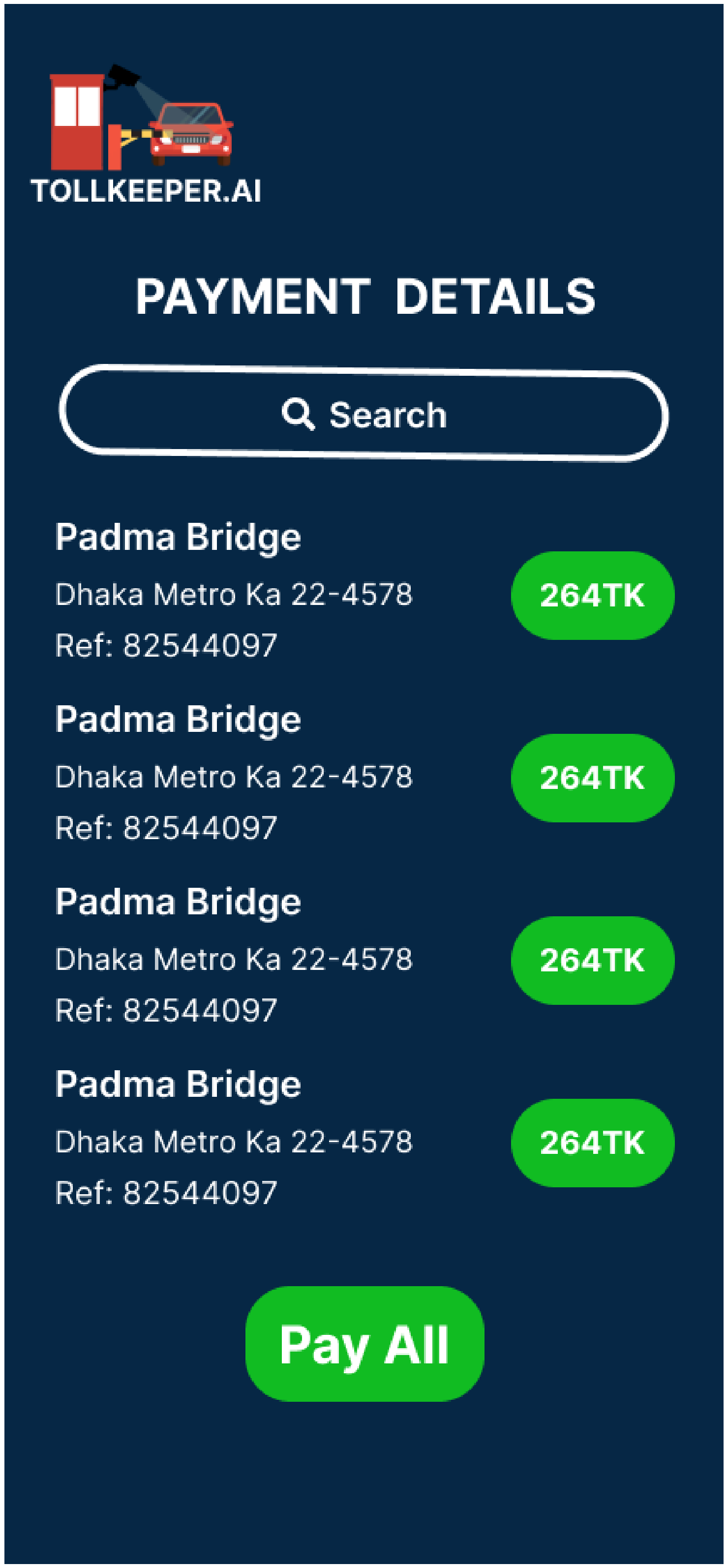
2. UX Designs



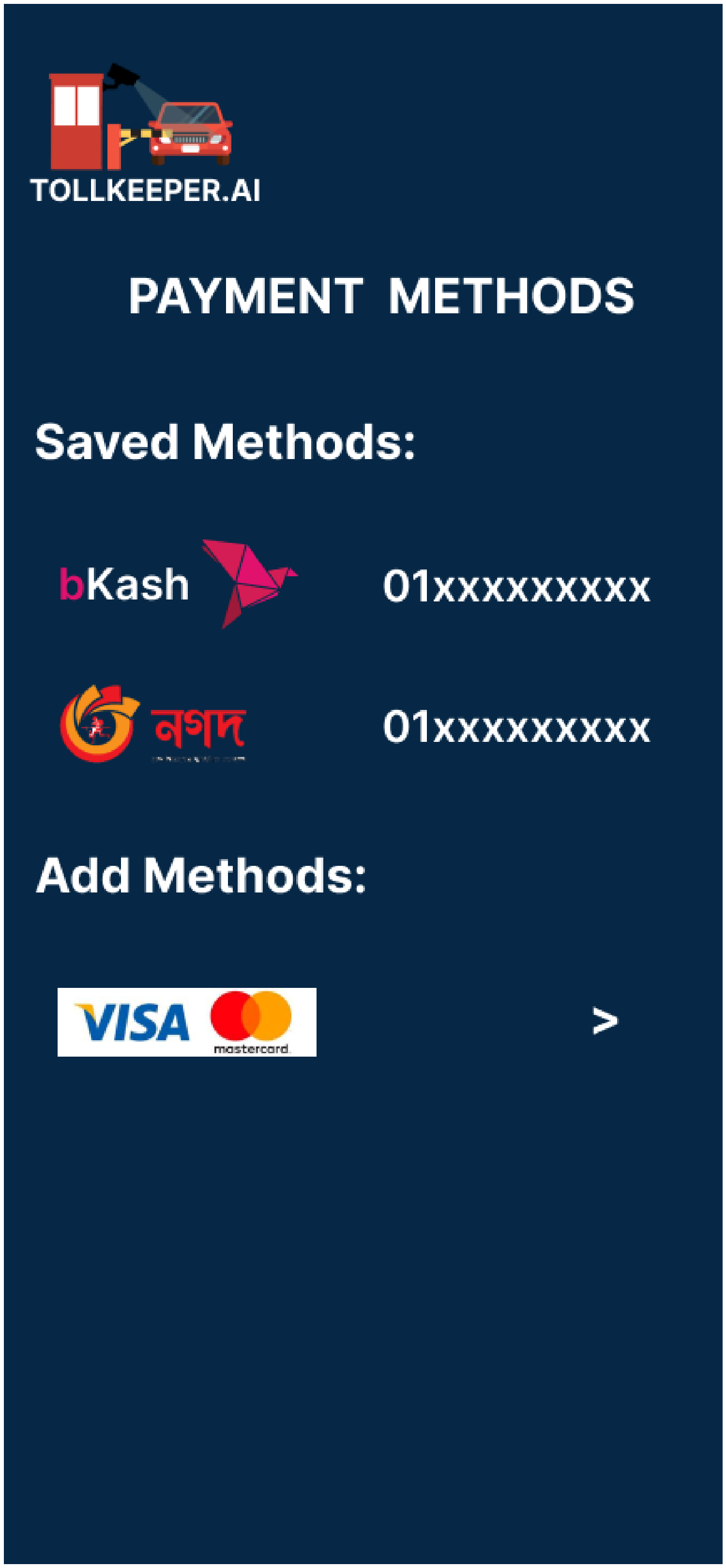
CO2: Design a system, component or process to meet desired needs within realistic constraints



CO2: Design a system, component or process to meet desired needs within realistic constraints

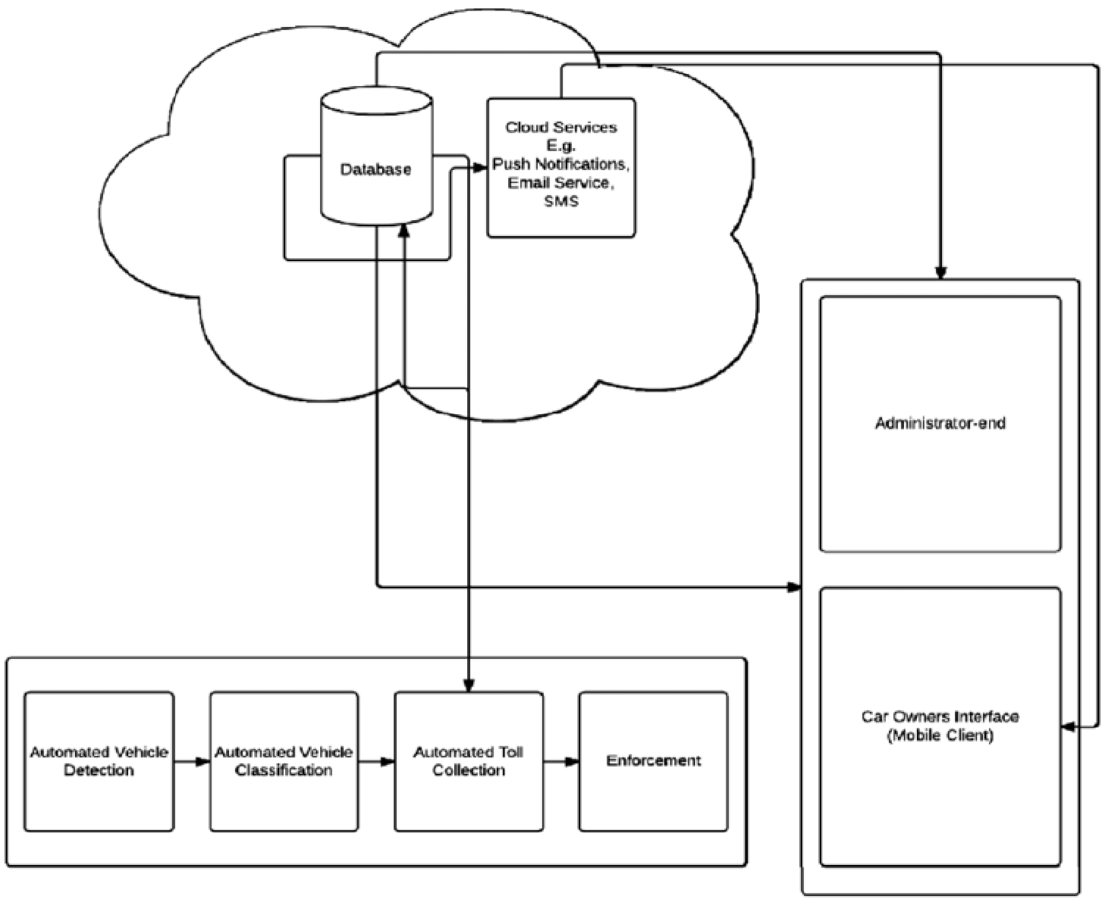


CO2: Design a system, component or process to meet desired needs within realistic constraints



CO2: Design a system, component or process to meet desired needs within realistic constraints

3. System Designs

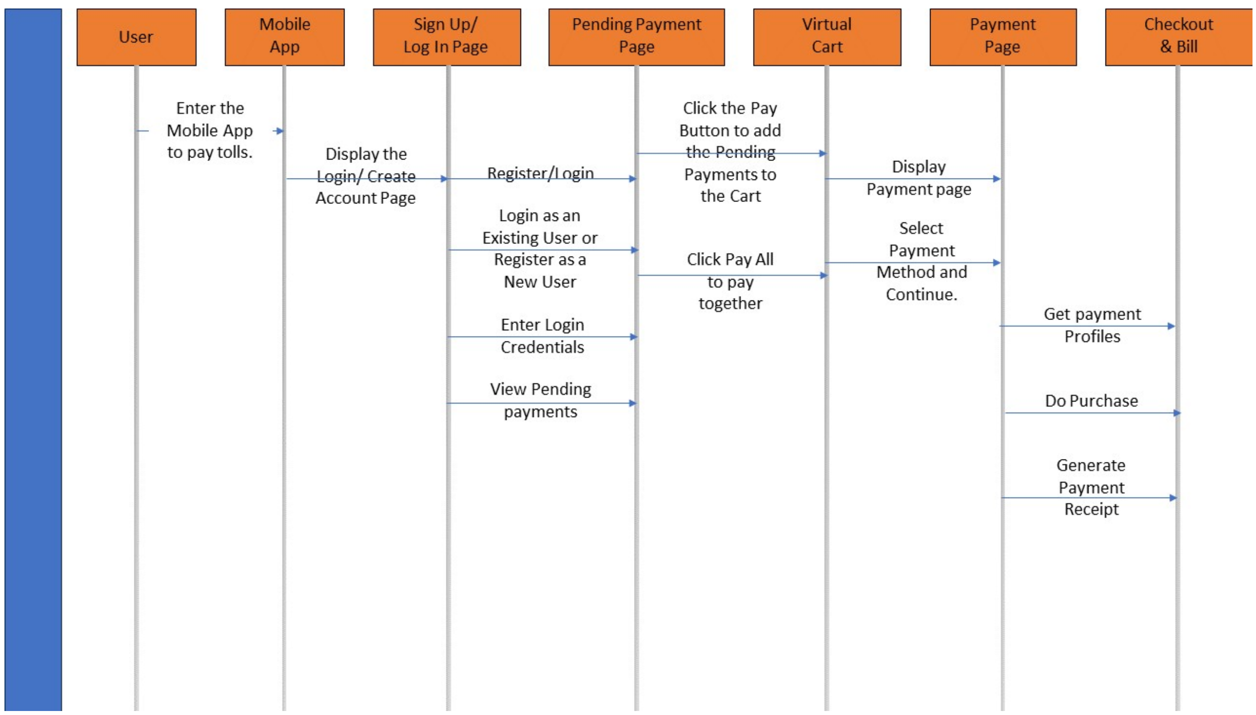


CO2: Design a system, component or process to meet desired needs within realistic constraints

4. Database design / Hardware Component Description

Hardware Components may include cameras and sensors.

5. Class diagram, sequence diagram, etc.



6. Relevant equations and algorithms

YOLO (You Only Look Once) is a deep network that uses residual and dense blocks to enable the flow of information to the deepest layers and to overcome the vanishing gradient problem. The YOLO object detection model mathematical equations and algorithms:

1. Grid and Anchor Boxes: YOLO divides the input image into a grid; typically, for each grid cell, it predicts bounding boxes. The number of bounding boxes expected per grid cell is determined by the number of anchor boxes used (usually 3 or 5). Each anchor box is associated with a predefined width and height, and the model predicts adjustments to these anchor boxes instead of absolute coordinates.
2. Predictions: For each bounding box, YOLO predicts Bounding Box Coordinates (x, y, w, h):
    - (x, y) represents the center of the bounding box relative to the grid cell.
    - (w, h) represents the width and height of the bounding box relative to the entire image.

These coordinates are transformed using sigmoid functions to ensure they fall in the range [0, 1]. Objectness Score: The objectness score indicates the probability that an object exists in the bounding box. The objectness score is also passed through a sigmoid function to constrain it within [0, 1]. Class Probabilities: YOLO predicts class probabilities for each bounding box. The class probabilities are calculated using a softmax function, ensuring they sum up to 1 across all classes.

3. Loss Function: YOLO uses a specialized loss function that combines errors in bounding box coordinates, objectness scores, and class predictions. The loss function penalizes incorrect predictions with varying weights for different components.
4. Non-Maximum Suppression (NMS): After predictions are made, YOLO applies Non-Maximum Suppression (NMS) to filter out redundant or overlapping bounding box predictions. NMS ensures that each object is detected only once by selecting the most confident bounding boxes and suppressing others.
5. Post-Processing: YOLO post-processes the bounding boxes after NMS by converting them from relative coordinates to absolute coordinates in the original image.
6. Output: The final output of YOLO is a list of detected objects, each represented by a bounding box with coordinates (x, y, w, h), a class label, and a confidence score.

The above description provides a high-level overview of YOLO's algorithm. In practice, YOLO uses multiple convolutional layers and other architectural details to make these predictions. The specific equations used in YOLO's loss function and architecture may vary between different versions of YOLO (e.g., YOLOv1, YOLOv2, YOLOv3, YOLOv4, etc.).