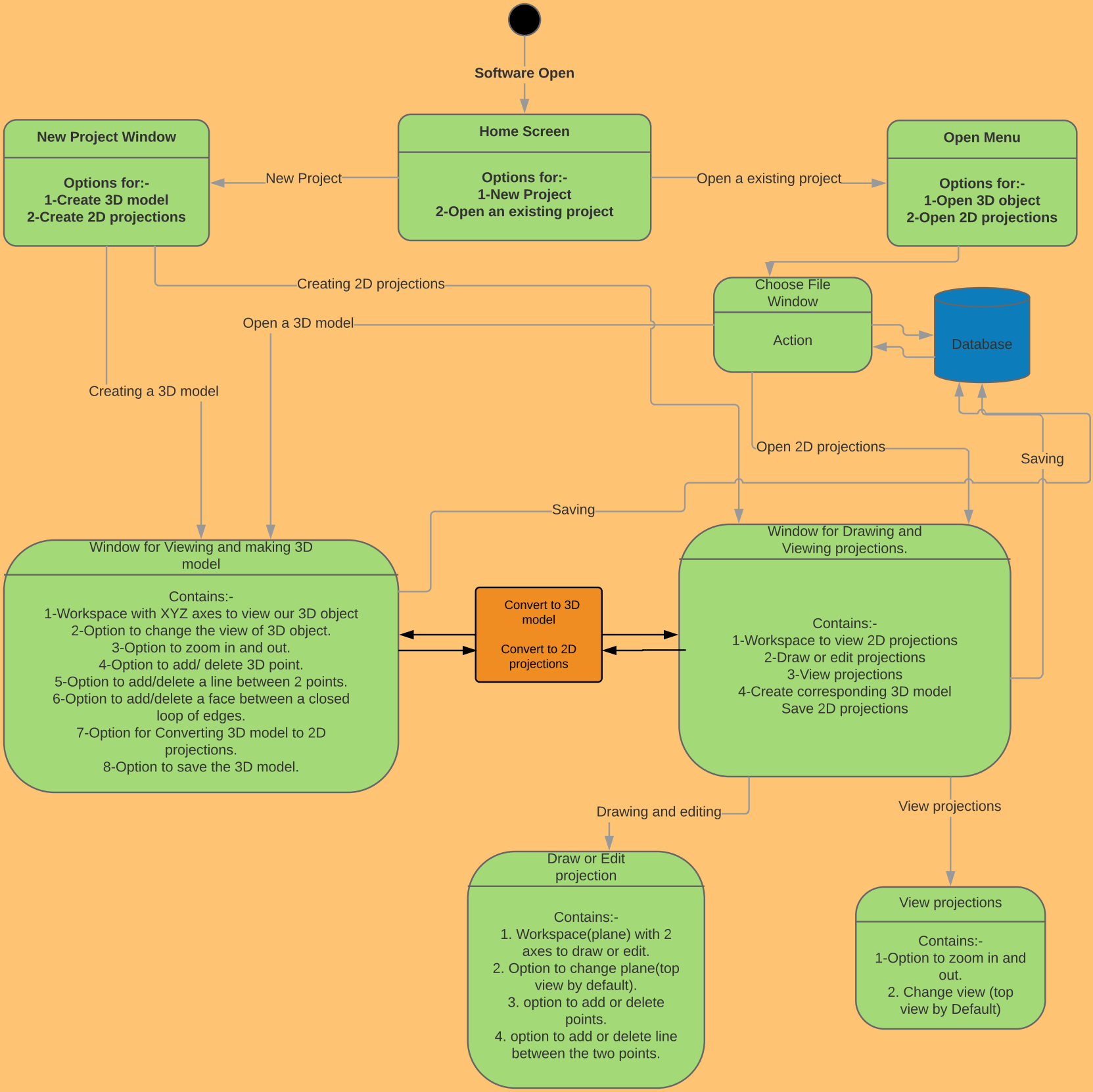
USER INTERFACE



Conversion of 3D model to 2D projections

Data Input

Initial data :-

We have data of 3D object
Input is of the form
1) List of all vertices of 3D object.
2)Data of which two vertices contain an edge
3) Data that which continous loop of edges contain a face

Module for Transferring Vertices in Projection:-

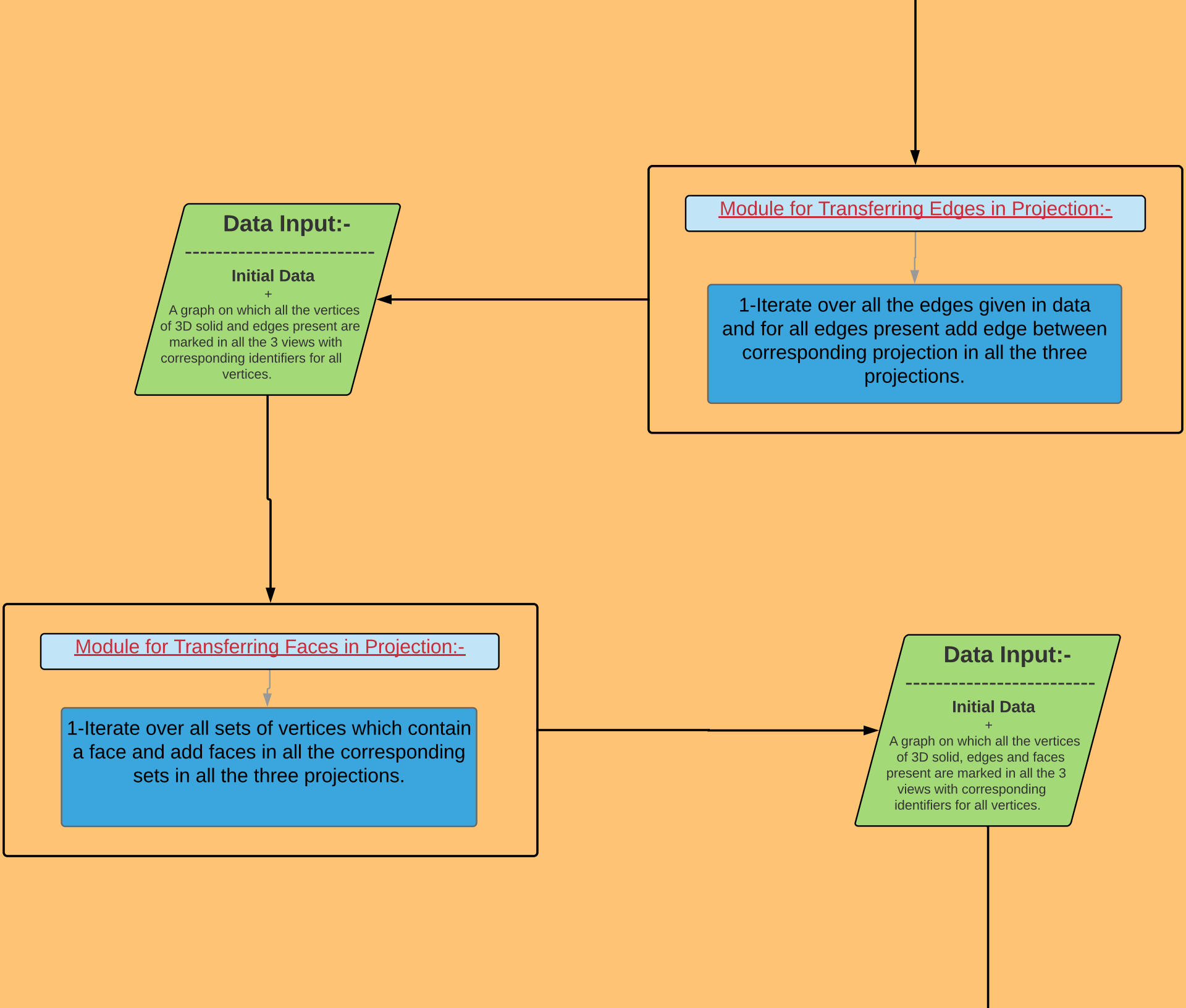
Top View Module:-
1-Iterate over all vertices and multiply by corresponding Rotation and Conversion Matrix
2- Mark each point with ' marker.

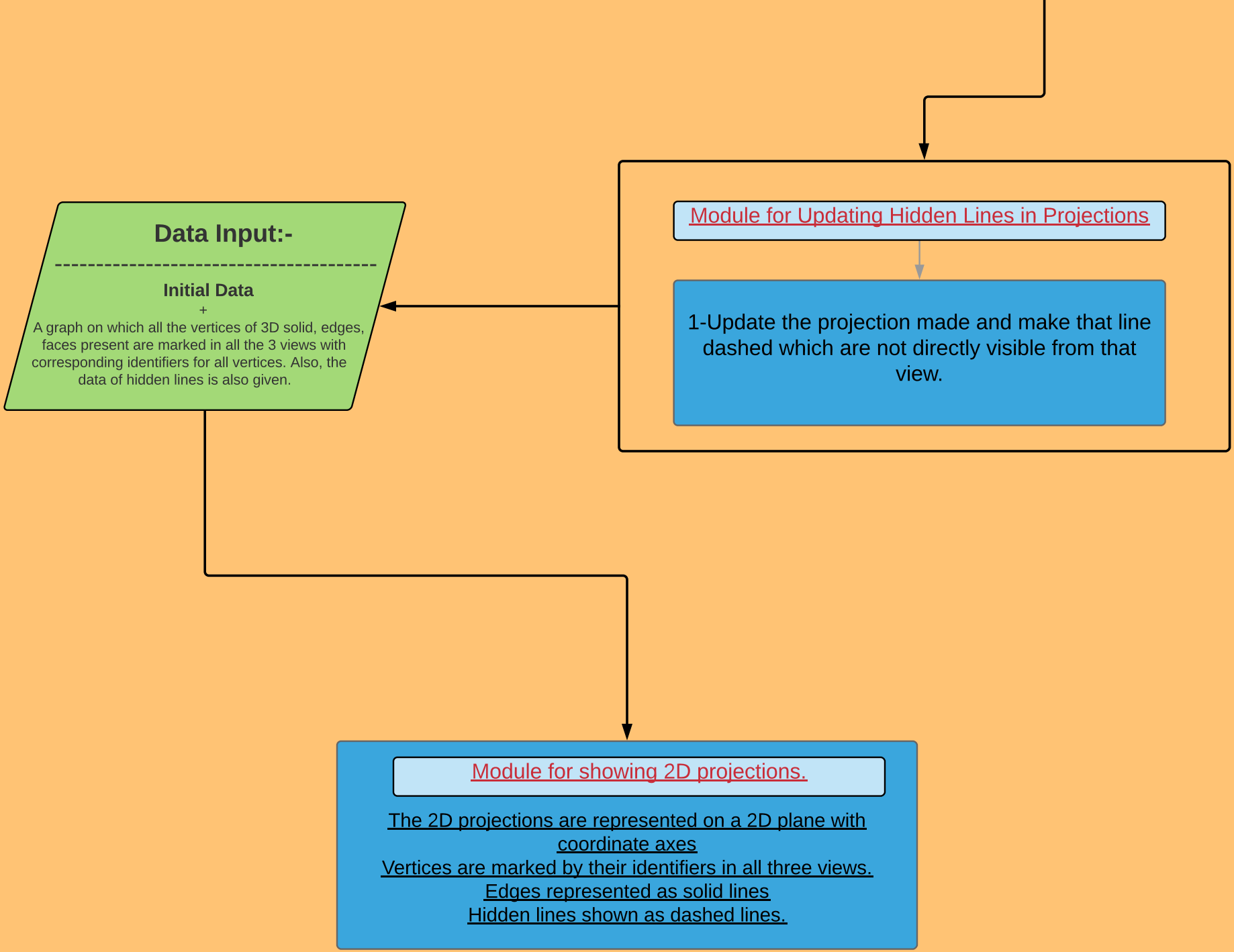
Front View Module:-
1-Iterate over all vertices and Multiply by its corresponding Conversion Matrix.
2-Mark each vertice with a unique indentifier.

Side View Module:-
1- Iterate over all vertices and mulitply by corresponding Rotation and Conversion Matrix.
2-Mark each point with '' marker

Data Input:-

Initial Data
+
A graph on which all the vertices of 3D solid are marked in all the 3 views with corresponding identifiers for all vertices


Module for Transferring Edges in Projection:-
1-Iterate over all the edges given in data and for all edges present add edge between corresponding projection in all the three projections.
Data Input:-
Initial Data
+
A graph on which all the vertices of 3D solid and edges present are marked in all the 3 views with corresponding identifiers for all vertices.
Module for Transferring Faces in Projection:-
1-Iterate over all sets of vertices which contain a face and add faces in all the corresponding sets in all the three projections.
Data Input:-
Initial Data
+
A graph on which all the vertices of 3D solid, edges and faces present are marked in all the 3 views with corresponding identifiers for all vertices.


Data Input:-

Initial Data
+
A graph on which all the vertices of 3D solid, edges, faces present are marked in all the 3 views with corresponding identifiers for all vertices. Also, the data of hidden lines is also given.
Module for Updating Hidden Lines in Projections
1-Update the projection made and make that line dashed which are not directly visible from that view.
Module for showing 2D projections.
The 2D projections are represented on a 2D plane with coordinate axes
Vertices are marked by their identifiers in all three views.
Edges represented as solid lines
Hidden lines shown as dashed lines.

Conversion of 2D projection to 3D model

2D PROJECTIONS

Initial data:
1 Pojection points(2D cordinates) for all the vertices (say P) (Points may be overlaping).
2. Line between these points (both hidden and clear) (say L).

Module for Vertex generator

From the given set of points we create a set of vertex:
When the three cordinates are equal in the three projection then the point is added to vertex set.

Set of vertices (say V) with there location in 3D space.

Module for Edge Generator

For each given pair in V check whether there is a line in set of lines L in all the projections (top, front and side view) if yes then add the pair in set of edges.

Set of edges (say E) with data about the pair of vertices they join. (may contain redundant edges)

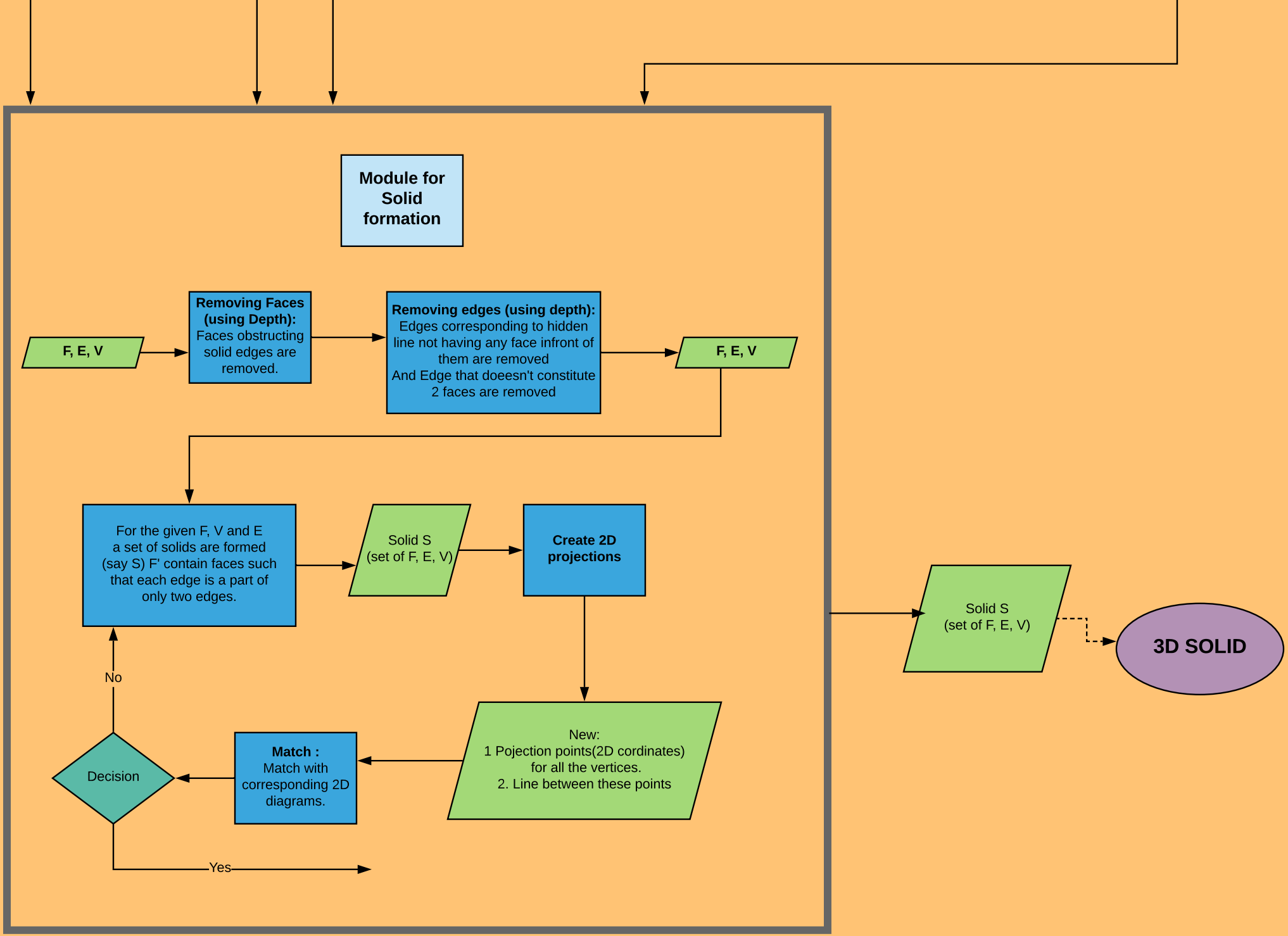
Module for Face Generator

Face formation:
1. Pick a vertex
2. Now check all the edges connected to vetex
3. Add those who lie on s, and their vertices are added to stack for further face formation and the process is repeated.

Surface Identifier:
Two edge sharing a common vertex lie constitute a surface.

Surface set (say s)

Set of faces(say F) withe the data about which edge loops forms the face. Some face might be reduntant


Module for Solid formation
F, E, V
Removing Faces (using Depth): Faces obstructing solid edges are removed.
Removing edges (using depth): Edges corresponding to hidden line not having any face infront of them are removed And Edge that doeesn't constitute 2 faces are removed
F, E, V
For the given F, V and E a set of solids are formed (say S) F' contain faces such that each edge is a part of only two edges.
Solid S (set of F, E, V)
Create 2D projections
New: 1 Pojection points(2D cordinates) for all the vertices. 2. Line between these points
Match : Match with corresponding 2D diagrams.
Decision
No
Yes
Solid S (set of F, E, V)
3D SOLID