System Architecture

Class/File Relationships and Flow:

```
1. Program.cs | |---> Initializes the main application window via:
| |---> MainForm.cs | |---> User interface for bus allocation
management (central hub). |---> Interacts with: | |--->
BusAllocator.cs |---> Retrieves data from Vars.cs, reformats for
the solver (Google OR-Tools). |---> Outputs allocation results into
the output box. | |---> DeptsCheckForm.cs |---> Opens a department
checklist. |---> Shows demand status for each department. |--->
Interacts with Vars.cs to retrieve demand data. |---> Calls
EditDemandsForm for editing department demands. | |--->
EditDemandsForm.cs |---> GUI for editing department demands. |--->
Interacts with Vars.cs for updating demand values. | |--->
SettingsForm.cs |---> GUI for editing app settings (e.g., bus
rates, buffer sizes, and file output settings). |---> Interacts
with Settings.cs to manage and store settings. | |---> IO.cs |--->
Handles input/output operations (reading/writing JSON files). |--->
Loads bus rates, route information, capacities, and demand data
from files. |---> Writes updated settings and demand data when
needed. | |---> Vars.cs |---> Central data repository that manages:
- TimeSets (route times). - Departments (and their demands). -
Routes (employee routes). - Capacities (bus capacities). |---> Used
by BusAllocator.cs, DeptsCheckForm.cs, and EditDemandsForm.cs. | |-
--> Settings.cs |---> Manages and stores application configuration.
|---> Interacts with IO.cs to load/save settings (e.g., bus rates,
file output settings).
```

Main Relationships:

1. **Program.cs**:
   - The entry point, initializing the `MainForm.cs`.
2. **MainForm.cs**:
   - Central hub for interactions with forms like `SettingsForm`, `DeptsCheckForm`, `EditDemandsForm`, and the `BusAllocator`.
   - Triggers bus allocations and displays output.

3. **BusAllocator.cs**:
    - Retrieves data from `Vars.cs`, uses the Google OR-Tools solver, and outputs the allocation results into the output log.
    - No longer exports allocations to a spreadsheet.
4. **Vars.cs**:
    - Acts as the central data manager. Stores and manages `Departments`, `TimeSets`, `Routes`, and `Capacities`.
    - Interacted with by the `BusAllocator`, `DeptsCheckForm`, and `EditDemandsForm`.
5. **IO.cs**:
    - Handles input/output for JSON and Excel files (bus rates, route information, demand data).
    - Interfaces with `Settings.cs` for saving/loading settings.
6. **SettingsForm.cs**:
    - Manages settings (bus rates, buffer sizes, allocation output folder) and interacts with `Settings.cs` and `IO.cs` to manage persistent configurations.
7. **DeptsCheckForm.cs**:
    - Provides the user interface for checking which departments have uploaded their demand data.
    - Calls `EditDemandsForm` for editing demands and interacts with `Vars` for current department data.
8. **EditDemandsForm.cs**:
    - Provides the interface to edit the demand data for each department.
    - Updates `Vars.cs` with the changes.

GUI Flow Overview:

1. **MainForm**:
    - The primary user interface where the user uploads bus rates, demands, sets dates, and generates allocations.
    - Buttons for accessing the settings form, clearing data, and generating allocations.

2. **SettingsForm**:
    - Allows toggling between "Individual Departments Mode" and "Total Demand Mode."
    - Options for editing buffer sizes and bus rates, as well as specifying the output folder for the allocations.
3. **DeptsCheckForm**:
    - Displays the status of department demands (whether filled or incomplete).
    - The user can edit demand data for each department by clicking the "Edit" button, which opens the `EditDemandsForm`.
4. **EditDemandsForm**:
    - Provides a table where demand numbers can be modified per route (IN and OUT times).
    - Allows filling empty values with zero and saving changes.