1 Internet Banking System

```
\1 Internet Banking System
```

```
Add your own dedication into PlantUML

For just $5 per month!
Details on https://plantuml.com/dedication

Welcome to PlantUML!

You can start with a simple UML Diagram like:

Bob->Alice: Hello

Or

class Example

You will find more information about PlantUML syntax on https://plantuml.com

(Details by typing license keyword)

PlantUML 1.2024.1beta1
[From string (line 2) ]

@startuml
!include https://raw.githubusercontent.com/adrianvlupu/C4-PlantUML/latest/C4_Container.puml
Cannot open URL
```

Level 2: Container diagram

Once you understand how your system fits in to the overall IT environment, a really useful next step is to zoom-in to the system boundary with a Container diagram. A "container" is something like a server-side web application, single-page application, desktop application, mobile app, database schema, file system, etc. Essentially, a container is a separately runnable/deployable unit (e.g. a separate process space) that executes code or stores data.

The Container diagram shows the high-level shape of the software architecture and how responsibilities are distributed across it. It also shows the major technology choices and how the containers communicate with one another. It's a simple, high-level technology focussed diagram that is useful for software developers and support/operations staff alike.

Scope: A single software system.

Primary elements: Containers within the software system in scope. Supporting elements: People and software systems directly connected to the containers.

Intended audience: Technical people inside and outside of the software development team; including software architects, developers and operations/support staff.

Notes: This diagram says nothing about deployment scenarios, clustering, replication, failover, etc.