

Online ticket purchase

```

sequenceDiagram
    participant User
    participant System

    User->>System: Start
    System->>User: Log in
    User->>System: Is information valid?
    System->>User: yes/no
    User->>System: Registration
    System->>User: Is information valid?
    System->>User: yes/no
    User->>System: Reserve ticket
    System->>User: Is ticket available?
    System->>User: no
    User->>System: Cancel reservation
    System->>User: Is date expired?
    System->>User: yes/no
    User->>System: Add to basket
    System->>User: Make payment
    System->>User: Is payment details valid?
    System->>User: yes/no
    User->>System: Choose number of tickets & delivery way
    System->>User: Send online version of ticket via mail/social media
    System->>User: Choose sales point
    System->>User: Notify & send the paper version of ticket to sales point
    System->>User: End
  
```

The diagram illustrates the online ticket purchase process, divided into two swimlanes: **USER** and **SYSTEM**.

USER Swimlane:

- Start:** The process begins with a start node (black circle) leading to the **Log in** use case.
- Log in:** The user logs in, leading to a decision diamond.
- Decision:** The user is either an **Old user** (leading to **Reserve ticket**) or a **New user** (leading to **Registration**).
- Registration:** The user registers, leading to a decision diamond.
- Decision:** The system checks if the **Is information valid?** (leading to **Update database** if yes, or back to **Registration** if no).
- Reserve ticket:** The user reserves a ticket, leading to a decision diamond.
- Decision:** The system checks if the **Is ticket available?** (leading to **End** if no, or **Add to basket** if yes).
- Add to basket:** The user adds the ticket to the basket, leading to a decision diamond.
- Decision:** The system checks if the **Is date expired?** (leading to **Refund buyer** if yes, or **Show error message** if no).
- Refund buyer:** The user receives a refund, leading to **Update database**.
- Show error message:** The user receives an error message, leading to **End**.
- Choose number of tickets & delivery way:** The user chooses the number of tickets and delivery way, leading to a decision diamond.
- Decision:** The user chooses **online** (leading to **Send online version of ticket via mail/social media**) or **sales point** (leading to **Choose sales point**).
- Choose sales point:** The user chooses a sales point, leading to **Notify & send the paper version of ticket to sales point**.
- End:** The process ends at a final node (black circle).

SYSTEM Swimlane:

- Log in:** The system provides the **Log in** use case.
- Is information valid?:** The system checks if the information is valid.
- Registration:** The system provides the **Registration** use case.
- Is information valid?:** The system checks if the information is valid.
- Reserve ticket:** The system provides the **Reserve ticket** use case.
- Is ticket available?:** The system checks if the ticket is available.
- Add to basket:** The system provides the **Add to basket** use case.
- Is date expired?:** The system checks if the date is expired.
- Refund buyer:** The system provides the **Refund buyer** use case.
- Show error message:** The system provides the **Show error message** use case.
- Choose number of tickets & delivery way:** The system provides the **Choose number of tickets & delivery way** use case.
- Send online version of ticket via mail/social media:** The system provides the **Send online version of ticket via mail/social media** use case.
- Choose sales point:** The system provides the **Choose sales point** use case.
- Notify & send the paper version of ticket to sales point:** The system provides the **Notify & send the paper version of ticket to sales point** use case.
- End:** The system ends at a final node (black circle).

