Movie Design Project Flowchart Blue: store method Purple: command Yellow: transaction method Helina Azer, Daniella Pairault, Divyashna Chandra, Vandhana Prabhakaran Green: customer method readMovie() readCustomers() readCommands() reads in the customer text file and then inserting through checks into the map of customer recrords. reads in the movie text file and then inserting through checks into the store vector of movies. then creates a queue of "commands" based on the type named transactions assumption: movies.txt three children; comedyFactory, classicFactory, and dramaFactory. It register assumption: commands.txt adds a movie in createCustomer() addCustomer() the vector of pointers to valid file or not empty Movies dramaFactory comedyFactory After the genre is read it After the genre is read it registers the genre and After the genre is read it registers the genre and creates the drama movie registers the genre and creates the classic movie object. invalid char creates the comedy movie valid char command creates a new creates a new drama creates a new omedy movie based movie based on the omedy movie based on the parameters arameters passed ir on the parameters assed in and adds it and adds it to the assed in and adds i to the store to the store store parseHistory() parseBorrow() parseReturn() parses string to enable a history object to be made. parses string to enable a borrow object to be made. create new create new create new Comedy object Drama object Drama object with details with details with details —invalid id√—valid id— ,—invalid id—√ —invalid id—∕-ivalid id throw error stomer/mov find() throw error throw error Does this object Does this this object takes in the proper ensures that the customer object already exis already exist in already exist in parameters to return a huistory object to push in transaction queue ensures that the in the store? the store? the store? loops until end of file _invalid movie人 Increase Increase Increase Add the Add the the stock of that movie by 1 the stock of the stock of movie to the store movie to movie to throw error throw error that movie that movie the store the store by 1 by 1 takes in the proper parameters to return a borrow object to push in transaction queue takes in the proper parameters to return a return object to push in transaction queue manage movie types —empty transaction queue— **Comedy Class** After the different movies are read, Note: This is purely Transactions informational on how transactions are manages Transactions class will hold the from transactions parsed information uncovered by readCommandFile(). By putting processCommands() the parsing load up front, it makes it easier to access the data when pushes to transaction queue we are performing the operations. operator< operator< operator to sort by title operator to sort by title loops until empty operator< operator== title is not equal operator to sort by operator to sort by release year createBorrow createInventory createHistory allows a customer to return while adjusting elements accordingly ouptputs the inventory of the store title is not equal title is not equal Borrow holds the customerID, the Return holds the customerID, the Inventory holds no data members vector of media types, and the vector of media types, and the because there is no additional History only holds the customerID operator< operator< vector of movie types. This makes vector of movie types. This makes data provided with this because that is all the information it possible to gather the search it possible to gather the search transaction. This class is simply to required for this function. operator to sort by year criteria to perform on the movies criteria to perform on the movies identify the type of transaction operator< operator to sort by year operator to sort by year end process